10/829,613 PTO-892

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 December 2002 (27.12.2002)

PCT

(10) International Publication Number
WO 02/103577 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US02/19541

(22) International Filing Date: 19 June 2002 (19.06.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/299,243 19 June 2001 (19.06.2001) US

(71) Applicant *(for all designated States except US)*: **POLYVISTA, INC** [US/US]; 1222 Ridgeley Drive, Houston, TX 77055 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **ANWAR, Mohammed, Shahbaz** [US/US]; 1222 Ridgeley Drive, Houston, TX 77055 (US). **DAHALE, Venkatesh** [IN/US]; 5801 Spring Valley #1008, Dallas, TX 75240 (US).

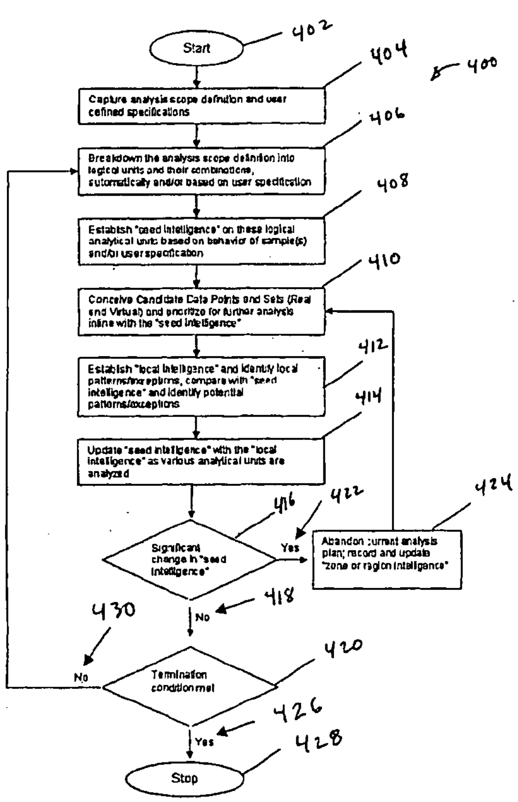
(74) Agent: **STROZIER, Robert, W.**; 2925 Briarpark Drive, Suite 930, Houston, TX 77042 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: A METHOD FOR EXCEPTIONS DETECTION IN N-DIMENSIONAL DATA SETS WITH FAST CONVERGENCE



(57) Abstract: A methodology is described for enhancing data mining processing using virtual database hierarchical constructs, that have dimensionality structure designed for improved data handling by data mining routine or algorithms. The methodology also includes static and/or dynamic data binning routines. The binning routines coupled with the virtual hierarchical constructs provide improved data anomaly detection and enhanced user directed query and data analysis functionality.

WO 02/103577 A2

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

TITLE: A METHOD FOR EXCEPTIONS DETECTION IN N-DIMENSIONAL DATA SETS WITH FAST CONVERGENCE

INVENTOR: Mohammed S. Anwar and Venkatesh Dahale

ASSIGNEE: PolyVista, Inc., a Corporation of the State of Delaware, United States

RELATED APPLICATIONS

This application claims provisional priority to United States Provisional Patent Application Serial No. 60/299,243 filed 19 June 2001.

BACKGROUND OF THE INVENTION

1. Field of Invention

[0001] The present invention relates to a method for detecting exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically and virtually generating data points based on base aggregated data to facilitate rapid data mining in large n-dimensional datasets, especially datasets associated with OLAP cubes.

[0002] More particularly, the present invention relates to method for detecting data anomalies, exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically and virtually generating data points based on base aggregated data including the steps of selecting at least one multi-dimensional dataset and at least one measure associated with the data variables; constructing a virtual (imaginary) database schema from a native database schema of the dataset to reduce the dimensionality of the data, while maintaining the measure or producing a Meta measure of more than one measure; selecting a limited number of data variables from the native schema of the dataset; creating an initial global rule describing the behavior of the measure with respect to the limited number of data variables; determining regions of data that would violate the initial global rule; selecting on of the regions; searching the dataset for data the falls within the selected region to form an exception dataset; and reporting the exception dataset.

2. Description of the Related Art

[0003] Computer technology continuous to allow the storage of ever increasing amount of data. This data storage explosion has given rise to new database technology for storing and retrieving data. However, the interfaces between the user and algorithms designed to delve into the data to find hidden relationships, anomalies, trends or the like is lagging far behind.

[0004] Large databases are usually aggregated and summarized along pre-defined, frequently

used taxonomy or hierarchy in Multi-Dimensional OLAP databases or in Relational databases to improve ad-hoc query performance in accessing aggregated information.

[0005] While such structures facilitates ad-hoc querying and cross-tab reporting, which remains the predominant objective for such systems, the search for exceptions and patterns usually remains limited along and across these pre-defined taxonomies and based on biased numeric thresholds of one or more measures considered individually or in limited combinations.

[0006] Many data mining techniques exist, which may be used to identify patterns or detect exceptions by processing data at a pre-defined level of granularity and across measure combination(s) (Analysis Context). However, such automated techniques suffer from the same drawbacks as the manual techniques, in that they lack intelligence to self determine Analysis Context and progress across and/or switch between Analysis Contexts, given an initial analysis scope (Search Path).

[0007] One way of improving pattern identification and exception detection is to device a way to overcome "Hard Boundaries" imposed by pre-aggregated structures, while utilizing the aggregated values where possible and by making the exception and pattern detection methods intelligent enough to self specify Analysis Context and Search Path, while providing provisions to integrate human intelligence.

[0008] Thus, there is a need in the art for a methodology for increasing data mining of the data stored in databases, especially in the area of anomaly identification.

SUMMARY OF THE INVENTION

[0009] The present invention provides a method implemented on a computer to alter a dimensionality of a multi-dimensional database hierarchical structure, iteratively and dynamically, to enhance, increase and/or make more efficient data processing so that qualified data points are made available to various data mining algorithms. The virtual (imaginary) alteration of the dimensionality of the database structure can be to increase or decrease the dimensionality of any part of the database structure of interest, *i.e.*, to alter the Dimensions, the Levels, the Members, and/or the Measures depending on the requirements of the data mining algorithm, the user query and/or the form of the information sought from the query. The qualified data points can be Crosstabs, Crossjoins, Meta Exceptions or the like. The data mining routine can be any routine that is designed to detect patterns and/or exceptions in multi-dimensional space; with an ability to specify/modify the dimensionality

and/or data point qualifications – inline with algorithm/process/ user requirements. In addition, the special constructs can also be utilized for user defined ad-hoc reporting.

[0010] The present invention provides a computer having stored thereon code of the methodology described above.

[0011] The present invention provides a computer readable medium having stored thereon code of the methodology described above.

[0012] The present invention also provides an analysis wizard records the initial definition of search universe and any user defined customization to the pre-aggregated structures, thus accommodating any existing aggregated structures without imposing any special structural requirements.

[0013] The method of this invention provides for initial routines that study the major behavior of a sample dataset allowing generation of a "seed intelligence" or global intelligence or rule describing the major behavior of the measure(s). The seed intelligence is used to create new virtual data points and/or crosstabs and to create Analysis Context along and across previously existing data points and virtual data points, which represent data regions which represent data values that would violate the seed intelligence and prioritizing of such candidate Analysis Contexts towards converging on anomaly or exception quickly. As the analysis progresses through the various candidate Analysis Contexts, the "Seed Intelligence" is constantly revised and the candidate Analysis Contexts are re-prioritized and/or revised. The virtual data point members are created by the algorithm based on both - the measure value thresholds, member ranges and qualified member lists in existing taxonomies. This helps in both - improving scalability and in fine tuning the convergence to anomaly or patterns quickly. Local behavior as well as revising overall behavior is used for detecting anomalies and pattern, exceptions and patterns are made available for user perusal as they are detected. Exceptions and patterns are presented within the prevailing Analysis Context in easy to understand form.

[0014] The present invention, thus, provides a method for detecting exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically and virtually generating data points based on base aggregated data to facilitate rapid data mining in large n-dimensional datasets, especially datasets associated with OLAP cubes.

[0015] The present invention also provides a method for detecting data anomalies, exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically

and virtually generating data points based on base aggregated data including the steps of selecting at least one multi-dimensional dataset, preferably in the form of an OLAP cube, and at least one measure associated with the data dimension in the dataset; capturing a scope of analysis and constraints from a user; constructing a virtual database schema from a native database schema of the dataset to reduce or expand the dimensionality of the dataset as a whole or in regions of interest, while maintaining the associated measure or producing a composite measure from more than one measure; selecting a limited number of data values from the entire dataset or the part of interest; creating an initial global rule, "seed intelligence" describing the behavior of the measure with respect to the selected, limited number of data values; determining data regions that would violate the initial global rule; prioritizing the regions; searching the dataset for data the satisfies the initial seed intelligence and that falls within the regions forming regional datasets; and reporting the regional datasets.

[0016] The present invention also provides a method for detecting data anomalies, exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically and virtually generating data points based on base aggregated data including the steps of selecting at least one multi-dimensional dataset, preferably in the form of an OLAP cube, and at least one measure associated with the data dimension in the dataset; capturing a scope of analysis and constraints from a user; constructing a virtual database schema from a native database schema of the dataset to reduce or expand the dimensionality of the dataset as a whole or in regions of interest, while maintaining the associated measure or producing a composite measure from more than one measure; selecting a limited number of data values from the entire dataset or the part of interest; creating an initial global rule, "seed intelligence" describing the behavior of the measure with respect to the selected, limited number of data values; determining data regions that would violate the initial global rule; prioritizing the regions; searching the dataset for data the satisfies the initial seed intelligence forming a compliance dataset and that falls within the regions forming regional exception datasets; if the regional exception datasets are not null (empty) or do not contain too few data points to support statistical analysis, creating regional intelligence or local intelligence; determining datapoints within each regional exception dataset that represent exceptions to the local intelligence; and reporting the results.

[0017] The present invention also provides a method for detecting data anomalies, exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically

and virtually generating data points based on base aggregated data including the steps of selecting at least one multi-dimensional dataset, preferably in the form of an OLAP cube, and at least one measure associated with the data dimension in the dataset; capturing a scope of analysis and constraints from a user; constructing a virtual database schema from a native database schema of the dataset to reduce or expand the dimensionality of the dataset as a whole or in regions of interest, while maintaining the associated measure or producing a composite measure from more than one measure; selecting a limited number of data values from the entire dataset or the part of interest; creating an initial global rule, "seed intelligence" describing the behavior of the measure with respect to the selected, limited number of data values; determining data regions that would violate the initial global rule; prioritizing the regions; searching the dataset for data the satisfies the initial seed intelligence forming a compliance dataset and that falls within the regions forming regional exception datasets; if the regional exception datasets are not null (empty) or do not contain too few data points to support statistical analysis, creating regional intelligence or local intelligence; determining datapoints within each regional exception dataset that represent exceptions to the local intelligence; update the initial seed intelligence with the local intelligences properly weighted to form an updated seed intelligence; comparing the updated seed intelligence; if the updated seed intelligence is significantly different from the initial seed intelligence, replacing the initial seed intelligence with the updated seed intelligence; repeating the previous three steps, until there is no significant change between the seed intelligence from the previous iteration and this iteration; and reporting the results. The method can also include a final test to determine whether a termination condition has been met, where failure to met the condition would restart the analysis construction of the scope of analysis step and the method steps would be continued until the condition is met.

[0018] The present invention also provides a method for constructing an intelligence models including an overall or global intelligence and local intelligences using the methods set forth above, which generates the intelligences from the analysis of data in multidimensional databases, relational or OLAP, and in the use the intelligence model to predict further data behavior.

[0019] The present invention also provides a method for constructing libraries of intelligence models, each model including an overall or global intelligence and local intelligences using the methods set forth above, which generates the intelligences from the analysis of data in

multidimensional databases, relational or OLAP.

[0020] The present invention also provides a method for using the library of intelligence models to classify data behavior and as a tool for predicting the behavior of classified data. and in the use the intelligence models to predict further data behavior.

[0021] The present invention provides a computer having stored thereon code corresponding toe the above provided methods.

[0022] The present invention provides a computer readable medium having stored thereon code corresponding toe the above provided methods.

DESCRIPTION OF THE DRAWINGS

[0023] The invention can be better understood with reference to the following detailed description together with the appended illustrative drawings in which like elements are numbered the same:

[0024] Figures 1A-D depict the structure of OLAP databases showing dimensions, measures and values and illustrating the formation of composite measure - dimensional reduction and illustrating the identification of exception and meta exception candidate data regions;

[0025] Figures 2A-D illustrate a wizard for defining analysis scope and identifying exception candidate data regions;

[0026] Figures 3A-D illustrate the on the fly binning process and the results derived therefrom;

[0027] Figure 4 depicts a conceptual flowchart of a preferred method of this invention, which illustrates an iterative method for detecting exceptions and patterns in a specified analysis scope through guided analysis of real and virtual data points and/or crosstabs in an OLAP cube;

[0028] Figure 5 shows a cube having defined with a dimensionality of four dimensions and one measure;

[0029] Figure 6 depicts the construction of a composite measure in the cube of Figure 4, the Performance Monitors (Metrics) are arranged in a dimension, the tuple (Member of Performance Monitor Dimension, Member of Measures Dimension) in accord with step **306** of Figure 3, a wizard helps in the specification and customization of any special structures;

[0030] Figure 7 shows an example of a wizard, which records user inputs related to a dimension level listing the measure metrics. The user selects the members, which will be studied as composite measures from the Performance Monitors dimension. The user selects

"Memory-Bytes Available" and "Memory-Pages Per Sec" members from the Performance Monitor dimension.

[0031] Figure 8 shows the next screen of the wizard, which allows for the specifying the measures that quantify the value of Performance Monitor members;

[0032] Figure 9 depicts a window showing the results of the operation of the wizard of Figure 7;

[0033] Figure 10 depicts the result of dimension reduction be concatenation of measures into the member dimensions;

[0034] Figure 11 depicts a screen showing the next step of the wizard operation, where a sample population is polled to allow construction of a seed intelligence or an initial guess of a global rule defining the relationship between the members being correlated;

[0035] Figure 12 depicts a widow showing the constructed relationship or seed intelligence from the sample population;

[0036] Figure 13 is a plot graphically depicting the seed intelligence as a straight line with negative slop, a grid binning the plot into nine bin valued regions, and the identification of exception regions shaded for the sake of highlighting;

[0037] Figure 14 is a plot graphically depicting a preferred method for determining local intelligence and local intelligence exceptions involving binning data points within the exception regions on a finer scale;

[0038] Figure 15 depicts a screen showing crosstab results from MDX code based on the binning process illustrated in Figure 14;

[0039] Figure 16 depicts a screen showing crosstab results from the iterative analysis, where global ("seed intelligence"), regional and local behavior of composite measure is determined and revised.;

[0040] Figure 17 depicts a screen showing crosstab results of a hybrid analysis;

[0041] Figure 18 depicts a binning configuration for testing the global intelligence, and identifying and testing local intelligence;

[0042] Figure 19 depicts an iterative processes for analyzing data points within each bin of Figure 18;

[0043] Figure 20 depicts a plot of the results of the iterative processes and the identification of an exception within the intelligences;

[0044] Figure 21A and Figure 21B depict a plot showing the seed intelligence and deviations

from the seed intelligence analyzed in a step wise binning process, where first Memory Paging binned slices are analyzed followed by Memory Availability binned slice analysis; and

[0045] Figure 22 depicts a screen of an analysis which has resulted in the confirmation of the seed intelligence.

DETAILED DESCRIPTION OF THE INVENTION

[0046] The inventors have developed a methodology for facilitating the identification of exceptions or anomalies in data via the construction of global rules from a sample selection of data in a multidimensional dataset and for the identification of regions of data that do not obey the rule (exceptions) and the construction of local rules to identify exceptions to the local rules. The inventors have found that this methodology has the following benefits and applications: (1) eliminates the need to create Binned Dimensions with Pre-Defined Intervals-Bins (Stored or Virtual) in OLAP cubes; (2) provides a new way of studying the interaction between Binned Variables in n-Dimensional Space; (3) provides a new way to detect anomalies (Dimensional Context associated with Data Bin Context) based on the concept of Exception and Meta-Exception - Tupled Bins with anomalous associated data; (4) provides a new way to converge on the Meta-Exceptions faster; (5) accommodates special cube structures and measures constructs. These data constructs, when organized in Categorical or Regular Dimensions, can be tupled with Measures to define Meta-Measure or Composite Measure along which the dimensional crosstabs can be reported and/or analyzed for exceptions.

[0047] The present invention relates broadly to a system for finding global and local data patterns and exceptions to both the global pattern and the local pattern, where the system includes an analysis scope capture and definition module, a breakdown module for breaking the analysis scope into logical units or combinations of logical units, a seed intelligence module that determines a seed intelligence (global rule) from a limited data selection from the data to be analyzed; a determine exception candidate region modules where regions of data which would violate the seed intelligence are identified, prioritized and analyzed inline with the analysis of the seed intelligence guess, a determine local intelligence and identify local intelligence exceptions and compare the local intelligence to the seed intelligence, a create an updated seed intelligence module, where the updated seed intelligence and test the updated seed intelligence against the current seed intelligence and repeat the analysis until

the updated seed intelligence and current seed intelligence differ by only an insignificant amount.

[0048] The present invention also broadly relates to a method for finding global and local intelligences quickly including the steps of capturing an analysis scope, a breakdowning the analysis scope into one or more logical units or combinations thereof, optionally specifying constraints on the analysis scope; establishing a seed intelligence from a sample data population, from user input or a combination of data sampling and user input, identifying data regions that represent exceptions to the seed intelligence, establishing local intelligence in each exception region, if non empty, updating seed intelligence with local intelligences or forming a composite intelligence of an updated seed intelligence and local intelligences; testing to determine if the seed intelligence or composite intelligence from the last cycle is significantly different than the seed intelligence or composite intelligence of this cycle, exiting changes are insignificant or returning to the identifying step if significant changes occurred for iteration until convergence is achieved. The method can also include a termination test. After convergence, the method will have constructed a consistent intelligence, seed or composite, for describing the data behavior and will have identified exceptional regions, local intelligence associated with the regions and exceptions to the local rules.

[0049] The present invention also relates broadly to the construction of intelligences, seed, local and/or composite, for the construction of model for predicting data behavior. The intelligences can also be pooled into a library for even faster trend and pattern analysis of n-dimensional datasets.

[0050] The methodology of the present invention is ideally suited for finding data exceptions, global and local intelligences or data patterns and composite intelligences or data patterns - mixtures of global and local data patterns or intelligences in data with many dimensions contained in any type of database, but preferentially contained within an OLAP database. The methods of this invention are ideally suited for the analysis of any type of multidimensional data including, without limitation, operational data, manufacturing data, financial data, currency exchange data, human behavioral data, medical data, regulatory data, legal data, or any other data that have many dimensions (members) and at least one measure (value).

[0051] The methods of this invention allow dimensional manipulations of the original database schema without having to change the original database schema. Thus, the method

creation of expanded or reduceed database schema to construct database schemas that do not correspond to the physical database schema of the databases being analyzed. These computer constructed database schema can be generated by the method, specified by the user or any combination of method generated and user specified constructions. The imagined or constructed database are designed to improve the efficiency and speed in generating global and local data patterns - intelligences, through an iterative or recursive method to refine the intelligences until they are self-consistent – do not vary significantly from one cycle to the next. The meaning of significant, of course, may change from analysis to analysis and may even be user defined. However, the term generally means that intelligences from two consecutive cycles differ by less than about 20% at each point along the graphical or mathematical representation of the intelligence, global, local and/or composite, preferably, the difference is less than about 15% at each point, particularly, less than about 10% at each point, and more particularly, less than about 5% at each point, with an ultimate goal being less than 1% at each point. Of course, the smaller the acceptable different, the longer the process will take to converge. Thus, for gross analysis, a 20% or greater difference may be acceptable; while for a detailed analysis, 1% or less may be acceptable.

[0052] This invention also relates to a method for automatically, interactively and dynamically generating candidate virtual data points and selecting real data points (Crosstabs, Crossjoins, Meta Measures and Meta Exceptions) in OLAP and RDBMS databases for a specified analysis scope; *a priori* and *post-priori* application of statistical, data mining techniques to prune and/or prioritize candidate data points per the analysis objective; application of statistical and data mining techniques to identify exceptions in or patterns across candidate data points and potentially, interactively revise candidate data point generation definition and prioritization to fine tune and expedite exception or pattern detection.

[0053] The method can also include generating candidate virtual data points includes virtual alteration of dimensionality (Splitting or Merging Dimensions, Levels, Members, Measures in the multi-dimensional database, Columns and Rows of Tables in Relational Databases. The method can also include generating candidate virtual data points is based on analysis definition and user defined, algorithm defined –static and/or dynamic thresholds and context. The method can also include generating candidate virtual and real data points are prioritized for and interactively subjected to statistical and data mining routines to detect exceptions and identify patterns. The method can also include detecting exceptions and patterns are

presented in the context of combination of real and virtual data points, which includes prevailing thresholds and conditions. The method can also include combining real and virtual data points are utilized for user defined ad-hoc reporting or analysis.

[0054] This invention also relates to a method to alter dimensionality (increase/decrease) (Dimensions, Levels, Members, Measures) of the multi-dimensional database, interactively and dynamically, such that qualified data points (Crosstabs, Crossjoins, Meta Exceptions) are made available to various algorithms (that detect patterns and/or exceptions in multi-dimensional space; with an ability to specify/modify the dimensionality and/or data point qualifications – inline with algorithm/process/ user requirements) for efficient processing. In addition, the special constructs can also be utilized for user defined ad-hoc reporting. An application of the above method for crosstab qualifications (based on user defined, algorithm defined - static or dynamic thresholds).

[0055] This invention also provide a method that utilizes the above two concepts to converge on multi-variant anomaly fast – Sizing the problem based on Meta-Exceptions, while simultaneously selecting viable/optimal candidates (Candidates can be one or more dimensions (and there combinations) and/or one or more members (and there aggregated combinations).

[0056] The methods of this invention present a unique way of displaying the detected results such that anomalies are presented in a Cause and Effect relationship that include various prevailing thresholds and conditions used by the algorithm.

<u>Introduction</u>

[0057] As an introduction to the methodology of this invention, in an OLAP cube, data lies at the intersection of dimensional members. Looking at Figure 1A, an OLAP cube having four dimensions is shown schematically in a display window **100**. The display window **100** includes columns **102a-d** for each of the four dimension: Dim A, Dim B, Dim C, and Dim D, a column **104** for the Dim Measure and a column **106** for the intersection value. The display window **100** also includes a header row **108** with headers boxes **110** and ten data rows **112**.

[0058] In an OLAP environment, data values are associated with intersections of dimension members and measures. Ad-Hoc reporting entails viewing data values that lie at the intersections of desired dimensional members. Ad-Hoc analysis entails viewing dimensional members that intersect to yield desired data value. Using Ad-Hoc analysis, the search for

unusual data or exceptional data values with respect to expected or predicted trend in the data values is described below.

Important Facts

[0059] An abnormal intersection value constitutes an exception to a general rule of expected or predicted data behavior. In the methodology of this invention, such exceptions can be applied as filters and can be defined by a composite of dimensional member constructs – a tuple of regular dimensional members and measures and conditional "Intersection Value." Looking at Figure 1B, a window **120** is shown that includes **Candidates for Exception**; while in Figure 1C, a window **122** is shown that includes **Candidates for Meta Exception,** where the difference between the window **120** and the window **122** lies in the reduction of the dimensionality by merging dimensions – Dim C + Dim D + Dim Measure —> Dim CDMeasure..

Sample MDX – Illustrating Meta Exception

[0060] For simplicity, consider the intersection defined by five dimensions as set forth in the following MDX (Multi-Dimensional Expression – a query language for OLAP databases) code:

```
WITH  SET [OLAPINTERSECTION] AS
'{NonEmptyCrossjoin(
            {[Customers].[Country].MEMBERS},
            {[Education Level].[Education Level].MEMBERS},
            {[Gender].[Gender].MEMBERS},
            {[Marital Status].[Marital Status].MEMBERS}
      )}'

MEMBER [MEASURES].[METAEXCEPTION1] AS
'   (   [   P   r   o   d   u   c   t   ]   .   [   A   l   l
Products].[Food].[Dairy].[Dairy],[Measures].[Unit Sales] )'
MEMBER [MEASURES].[METAEXCEPTION2] AS
'([Promotion Media].[All Media].[Bulk Mail],[Measures].[Sales
Count])'
SELECT
      FILTER([OLAPINTERSECTION],*/Intersections where Filter
      Conditions (1,2) are met/*
      (SUM({[OLAPINTERSECTION].CURRENTMEMBER},[MEASURES].[ME
TAEXCEPTION1])>80)
OR
(SUM
({[OLAPINTERSECTION].CURRENTMEMBER},[MEASURES].[METAEXCEPTIO
N2])< 8)
      )
ON COLUMNS
FROM SALES
```

[0061] Looking at Figure 1D, a display window **130** is shown including a sample meta-

exception result window, where Dim A is customers, Dim B is education, Dim C is gender, and Dim D is marital status, generated by the above MDX code is shown, where only non empty intersections corresponding to data values where the two Meta Exception rules are met.

[0062] Figures 2A-D illustrate the steps the methodology of this invention uses to specify a search for exceptional data. Looking at Figure 2A, a window is shown for selecting a crosstab measure type. Looking at Figure 2B, a window is shown for selection of transposed measure dimension "Performance Monitor." Looking at Figure 2C, a window is shown with the third step of the exception definition process, where the elements of the transposed measures dimension are displayed and selected. At this step, user constraints can be added between the measures and/or between the transposed measures dimension members. Looking at Figure 2D, a window is shown including the constructed measure and transposed measures dimension and child and descendent measure values in the crosstab columns next to the list of dimensions.

[0063] The MDX code generated and executed using the definition step described above is shown below:

```
WITH  SET [OLAPINTERSECTION] AS
'{NonEmptyCrossjoin(
      {[Hour].[Hour].MEMBERS},
      {[Configuration].[Configuration Name].MEMBERS},
      {[Computer].[Name].MEMBERS}
      )}'
MEMBER [MEASURES].[METAEXCEPTION1] AS'([Performance Monitor].[All
Performance Monitor].[Memory-Bytes Available],[Measures].[Sample
Avg])'
MEMBER [MEASURES].[METAEXCEPTION2] AS
'([Performance Monitor].[All Performance Monitor].[Memory-Pages
per Sec], [Measures].[Sample Avg])'

SELECT
FILTER([OLAPINTERSECTION],
      (SUM    ({[OLAPINTERSECTION].CURRENTMEMBER},
      [MEASURES].[METAEXCEPTION1] )>800000000)
      AND
      (SUM    ({[OLAPINTERSECTION].CURRENTMEMBER},
      [MEASURES].[METAEXCEPTION2] )< 800000000))
ON COLUMNS

FROM PERFORMANCE
```

[0064] The present invention also relies on the concept of data value binning to help simply facilitate exception identification and pattern construction. Binning can reduce the amount of data values to be analyzed and help to augment the local data behavior. The particular type of binning most useful in the application of this invention is so-called "on the fly binning."

On the Fly Binning – Based on Exception and Meta Exceptions Concepts

[0065] Binning can be defined as a process of mapping continuous values into categorical values or bins. A bin is a category, *e.g.*, a series of continuous values 1,2,3,4,5,6,7,8,9,10 can be binned to the following categorical valued bins 1 to 5 and 5 to 10. Binning adds a lot of value in the process of exception detection. Binning can amplify data effects, such that the previously diluted exceptions, which were hard to identify in the entire data population are now easily identified in segments of population (Profiling – Binned Members). Binning can also reduce the effort required for exception detection by providing a sampling approach (Sampling – Binned Sets).

[0066] In the OLAP world, the bins can be defined by the dimensional members and data values. The data values, Val(Tuple), can be binned into absolute or dynamic ranges to form complex bins For example, the binning can be as simple as:

```
([Measures].[Sales Count] ) > 5000 AND ([Measures].[Sales Count]
) < 5900 )
```

or more complex as:

```
([Customers].[All Customers].[Canada], [Education Level].[All
Education Level].[Bachelors Degree], [Gender].[All Gender].[M],
[Product].[All Products].[Drink].[Alcoholic Beverages].[Beer and
Wine].[Beer],[Measures].[Unit Sales])>
2([MEASURES].[METAEXCEPTION1])
AND
([Customers].[All Customers].[Canada], [Education Level].[All
Education Level].[Bachelors Degree], [Gender].[All Gender].[M],
[Product].[All Products].[Drink].[Alcoholic Beverages].[Beer and
Wine].[Beer],[Measures].[Unit Sales])<
3([MEASURES].[METAEXCEPTION1])
```

[0067] The bins can be based on equal count per bin, equi-count bins, user defined bins or dynamically set by using outlier identifiers such as standard deviation, average, median, mode, min, max or other statistical functions. As an example of definition and utilization of bins, consider the following MDX code:

```
SELECT
      ORDER({[Product].[Product Name].Members}, [Measures].[Unit
      Sales], BASC) ON ROWS, {[Measures].[Unit Sales],
      [Measures].[Profit],[Measures].[Sales Count]} ON COLUMNS
FROM SALES
```

[0068] The observed values for Unit Sales and Profit for the Products range over their represented data values in the dataset.

[0069] To sample/profile the Products based on Unit Sales and Profit to study Sales Count patterns, binning would be performed on the Products based on Unit Sales and Profit Values. For example, consider following MDX statements, which shows a sampling application:

```
WITH
```

```
        SET [ProductUnitSales_BIN1] AS 'FILTER({[Product].[Product
        Name].Members},ISNULL([Measures].[Unit Sales]))'
        SET [ProductUnitSales_BIN2] AS 'FILTER({[Product].[Product
        Name].Members},  (([Measures].[Unit  Sales]  >  0)  AND
        ([Measures].[Unit Sales] < 100)))'
    SELECT
        UNION({[ProductUnitSales_BIN1]} , {[ProductUnitSales_BIN2]
        }) ON ROWS, {[Measures].[Unit Sales], [Measures].[Profit],
        [Measures].[Sales Count]} ON COLUMNS
FROM SALES
```

[0070] The result of the above operation shows that we can apply the bins in terms of sets, as a bin range may return multiple products in a profile. The operation also shows that multiple bins can be unioned together to form sets of larger bin.

```
    WITH
        SET [ProductUnitSales_BIN2] AS 'FILTER({[Product].[Product
        Name].Members},  (([Measures].[Unit  Sales]  >  0)  AND
        ([Measures].[Unit Sales] < 130)))'
        SET        [ProductUnitSales_BIN3]        AS
        'FILTER({[Product].[Product    Name].Members},    ((
        [Measures].[Profit]> 50) AND ([Measures].[Profit]< 80)))'
    SELECT
        UNION({[ProductUnitSales_BIN2]} , {[ProductUnitSales_BIN3]
        }) ON ROWS, {[Measures].[Unit Sales], [Measures].[Profit],
        [Measures].[Sales Count]} ON COLUMNS
    FROM SALES
```

[0071] The results from the above MDX code presents Products that belong to Bin2 or Bin3 constructed using the UNION function. Alternatively, using the INTERSECT function, the code would present Products that belong to both Bin2 and Bin3

[0072] The method of this invention allows binning without having to key in the MDX code manually. For the data set forth above, the method would bin the Unit Sales and Sales Count at the Relational Level, then build the Bin Dimensions in OLAP cube and finally select the appropriate range dimension in the Dimension Tree. The present invention allows bins to be based on MetaException concepts, which permits binning based exclusively along particular Dimensions and/or Level members.

[0073] One of the application of "On The Fly Binning" as in the above example would be to sample the OLAP data, which can then be subject to further analysis. For example, consider following MDX statements, which show profiling application:

```
    WITH
        Member     [Product].[ProductUnitSales_BIN1]     as
        'AGGREGATE({FILTER({[Product].[Product Name].Members}, ((
        [Measures].[Profit]> 50) AND ([Measures].[Unit Sales] <
        100))})'
        Member     [Product].[ProductUnitSales_BIN2]     as
        'AGGREGATE({FILTER({[Product].[Product Name].Members}, ((
        [Measures].[Profit]> 50) AND ([Measures].[Profit]< 53))})'
    SELECT
```

```
        {     [Product].[ProductUnitSales_BIN1],
        [Product].[ProductUnitSales_BIN2]}) ON ROWS,
        {[Measures].[Unit    Sales],    [Measures].[Profit],
        [Measures].[Sales Count]} ON COLUMNS
    FROM SALES
```

[0074] The results of the above MDX codes presents two new members which are aggregates of the specified filter condition. Such members can be used to magnify/concentrate exceptions/patterns that would previously be less visible. Note that each of the bins represents is a Unique Member, so there is no need to use UNION or INTERSECT functions. For example, consider following MDX statements, which show another profiling application:

```
WITH
    Member    [Product].[ProductUnitSales_BIN1]    as
    'AGGREGATE({FILTER({[Product].[Product   Name].Members},   ((
    [Measures].[Profit]> 50) AND ([Measures].[Unit Sales] < 100)))})'
    Member    [Product].[ProductUnitSales_BIN2]    as
    'AGGREGATE({FILTER({[Product].[Product   Name].Members},   ((
    [Measures].[Profit]> 50) AND ([Measures].[Profit]< 53)))})'
    Member    [Customers].[CustomersUnitSales_BIN3]    as
    'AGGREGATE({FILTER({[Customers].[City].Members},   ((
    [Measures].[Profit]> 50) AND ([Measures].[Profit]< 100)))})'
    Member    [Customers].[CustomersSalesAverage_BIN4]    as
    'AGGREGATE({FILTER({[Customers].[City].Members}, (( [Measures].[Sales
    Average]> 0) AND ([Measures].[Sales Average]<'500)))})'
SELECT
    {[Product].[ProductUnitSales_BIN1],[Product].[ProductUnitSales_BIN2]}
ON ROWS,
    {[Customers].[CustomersUnitSales_BIN3], [CustomersSalesAverage_BIN4]}ON
    COLUMNS
FROM SALES
WHERE [Measures].[Unit Sales]
```

[0075] The result of this MDX code is displayed Figure 3A, which shows the values for the 2X2 cross tab with ProductUnitSale_Bin1 and 2 on the rows and CustomersUnitSales-Bin3 and 4 on the columns.

[0076] The above MDX results in a crosstab that shows bins along a particular axis based on separate measure values and that the concept could be utilized towards mere ad-hoc report (which is a less significant application) to complex bin permutations.

[0077] Now, consider some binning examples using a tuple of two or more dimensions:

```
    WITH
        Member    [Product].[ProductStoreTypeUnitSales_BIN1]    as
        'AGGREGATE({FILTER(NONEMPTYCROSSJOIN({[Product].[Product
        Category].Members}, {[Store Type].[Store Type].Members}),
        (( [Measures].[Profit]> 0) AND ([Measures].[Unit Sales] <
        100)))})'
        Member  [Product].[ProductStoreTypeSalesAverage_BIN2]  as
        'AGGREGATE({FILTER(NONEMPTYCROSSJOIN({[Product].[Product
        Category].Members}, {[Store Type].[Store Type].Members}),
        (( [Measures].[Sales Average]> 0) AND ([Measures].[Sales
        Average]<75 )))})'
```

```
      Member [Customers].[CustomersEducationUnitSales_BIN3] as
      'AGGREGATE({FILTER(NONEMPTYCROSSJOIN({[Customers].[State
      Province].Members},{[Education Level].[Education
      Level].Members}), (( [Measures].[Profit]> 0) AND
      ([Measures].[Profit]< 100)))})'
      Member [Customers].[CustomersEducationStoreCost_BIN4] as
      'AGGREGATE({FILTER(NONEMPTYCROSSJOIN({[Customers].[State
      Province].Members},{[Education Level].[Education
      Level].Members}), (( [Measures].[Store Cost]> 0) AND
      ([Measures].[Store Cost]< 500)))})'
SELECT
      {[Product].[ProductStoreTypeUnitSales_BIN1],[Product].[Pr
      oductStoreTypeSalesAverage_BIN2]} ON ROWS,
      {[Customers].[CustomersEducationUnitSales_BIN3],
      [Customers].[CustomersEducationStoreCost_BIN4] } ON COLUMNS
FROM SALES
WHERE [Measures].[Unit Sales]
```

[0078] The result of this MDX code is displayed Figure 3B, which shows the values for the 2X2 cross tab with ProductStoreTypeUnitSale_Bin1 and ProductStoreTypeSaleAverage_Bin2 on the rows and CustomersEducationUnitSales_Bin3 and CustomersEducationStoreCose_Bin4 on the columns and showing boxed crosstab member of interest.

[0079] The above Binning (Profiling) operation can be made available to other algorithms that work in multi-dimensional space, resulting in the detection of the highlighted cell as exceptional. The method can create more advanced bins by using multidimensional tuples and by using dynamic ranges. Binned Members have significant overhead as aggregations are calculated on the fly; however, the flexibility and the analytical enhancement offsets the increased computational overhead.

[0080] Although the Bins may not benefit from existing aggregations, aggregations can be flexibly created. This method is more efficient than a relational environment and more flexible than a pure OLAP. For example, consider the following simple Bins along Customer, Product and Promotion Dimensions as viewed along the Customer Dimension:

```
Member [Customers].[Customers_BIN1] as
'AGGREGATE({FILTER({[Customers].[City].Members}, (([Measures].[Unit
Sales]>0) AND ([Measures].[Unit Sales]< 50)))})', SOLVE_ORDER = 2
Member [Customers].[Customers_BIN2] as
'AGGREGATE({FILTER({[Customers].[City].Members}, (([Measures].[Unit
Sales]>= 50) AND ([Measures].[Unit Sales]< 100)))})', SOLVE_ORDER = 2
```

as view along the Product Dimension:

```
Member [Product].[Product_BIN1] as
'AGGREGATE({FILTER({[Product].[Product Category].Members}, ((
[Measures].[Unit Sales]> 0) AND ([Measures].[Unit Sales]< 100)))})',
SOLVE_ORDER = 1
Member [Product].[Product_BIN2] as
'AGGREGATE({FILTER({[Product].[Product Category].Members}, ((
[Measures].[Unit Sales]>= 100) AND ([Measures].[Unit Sales]< 150)))})',
SOLVE_ORDER = 1
```

and finally as viewed along the Promotion Dimension:

```
Member [Promotions].[Promotion_BIN1] as
'AGGREGATE({FILTER({[Promotions].[Promotion Name].Members},
(([Measures].[Unit Sales]> 0) AND ([Measures].[Unit Sales]< 100)))})',
SOLVE_ORDER = 3
Member [Promotions].[Promotion_BIN2] as
'AGGREGATE({FILTER({[Promotions].[Promotion Name].Members},
(([Measures].[Unit Sales]>= 100) AND ([Measures].[Unit Sales]<
200)))})', SOLVE_ORDER =3
```

[0081] Once the bins are formed, the combination of bins can easily be determined that yield high values – just by getting a crossjoined crosstab:

```
WITH
     Member [Product].[Product_BIN1] as
     'AGGREGATE({FILTER({[Product].[Product Category].Members},
     (( [Measures].[Unit Sales]> 0) AND ([Measures].[Unit
     Sales]< 100)))})', SOLVE_ORDER = 1
Member [Product].[Product_BIN2] as
'AGGREGATE({FILTER({[Product].[Product Category].Members}, ((
[Measures].[Unit Sales]>= 100) AND ([Measures].[Unit Sales]< 150)))})',
SOLVE_ORDER = 1
Member [Customers].[Customers_BIN1] as
'AGGREGATE({FILTER({[Customers].[City].Members}, (([Measures].[Unit
Sales]>0) AND ([Measures].[Unit Sales]< 50)))})', SOLVE_ORDER = 2
Member [Customers].[Customers_BIN2] as
'AGGREGATE({FILTER({[Customers].[City].Members}, (([Measures].[Unit
Sales]>= 50) AND ([Measures].[Unit Sales]< 100)))})', SOLVE_ORDER = 2
Member [Promotions].[Promotion_BIN1] as
'AGGREGATE({FILTER({[Promotions].[Promotion Name].Members},
(([Measures].[Unit Sales]> 0) AND ([Measures].[Unit Sales]< 100)))})',
SOLVE_ORDER = 3
Member [Promotions].[Promotion_BIN2] as
'AGGREGATE({FILTER({[Promotions].[Promotion Name].Members},
(([Measures].[Unit Sales]>= 100) AND ([Measures].[Unit Sales]<
200)))})', SOLVE_ORDER =3
SELECT
     CROSSJOIN ({[Customers].[Customers_BIN1],
     [Customers].[Customers_BIN2] }, {[Product].[Product_BIN1],
     [Product].[Product_BIN2]}) ON COLUMNS,
{[Promotions].[Promotion_BIN1], [Promotions].[Promotion_BIN2]} On ROWS
FROM SALES
WHERE [Measures].[Unit Sales]
```

[0082] The result of this MDX code is displayed Figure 3C, which shows the values for the 2X2 cross tab with Customer_Bin1 and 2 on the outer rows and their levels Product_Bin1 and 2 on the inner rows and Promotion_Bin1 and Promotion_Bin2 on the columns and showing boxed crosstab member of interest.

[0083] Evaluating this in a single crosstab, however, requires a lot of wait time and results in further complications because of Solve Order. While this can be used for Ad-Hoc reporting alone, the method of this invention preferably uses a Permutation based algorithm, which combines Binned Members from various dimensions and evaluates the combinations that yield

high/low values. Besides just High/Low combinations, these bins can be used as Composite Cases for Market Basket Analysis.

Conceptual Flowcart Illustration of a Preferred Method

[0084] Referring now to Figure 4, a conceptualization **400** of a preferred method of this invention is shown to include a start step **402**, which transfers control to a capture step **404**, where an analysis scope and user specifications are defined. After the analysis is defined, the defined analysis is broken down into logical units or combination of logical units in a breakdown step **406**. The breakdown can be performed automatically, can be based on user specifications or can be a combination of automated breakdown and user defined breakdown or constraints thereon.

[0085] After the defined analysis is broken down, a seed intelligence is generated for the logical units based on the behavior of a data sample, or is defined by the user or is generated and then modified by the user in an establish seed intelligence step **408**, where seed intelligence represent a guess at a global rule describing the behavior of the data within the sample. Next, the method **400** identifies and/or constructs and prioritizes data regions that would represent exceptions to or violations of the seed intelligence in a conceive candidate regions step **410**. Once the regions have been identified, continued analysis includes identifying data that satisfies the seed intelligence and data that falls within the regions proceeds simultaneously or inline.

[0086] Once the exception regions have been constructed and/or identified, data are then collected that satisfy the seed intelligence and the exception regions, if sufficient data is found in an exception regions, then local intelligence is generated and compared to the seed intelligence in an establish local intelligence step **412**. The local intelligence step **412** also identifies data exceptions to the generated local intelligences. After local intelligence analysis and construction and seed intelligence analysis, the seed intelligence is updated with respect both to the data consistence with the former seed intelligence and with respect to the local intelligences relative to a strength of the local intelligences in a update seed intelligence step **414** to produce an updated seed intelligence.

[0087] After the updating, the former and the update seed intelligences are compared in a conditional step **416** to determine whether a significant change in the seed intelligence has occurred. If no significant changes to the former seed intelligence has occurred in this cycle, then control is transferred along a NO branch **418** to a termination test step **420**; otherwise control is transferred along a YES branch **422** to an abandon current analysis step **424**, where

the former seed intelligence is replace with the updated and control is then transferred back to the conceive candidate regions step **410**. The termination step **420** check the termination condition and if its is met control is transferred along a YES branch **426** to a stop step **428**, where the results of the analysis are reported; otherwise control is transferred along a NO branch **430** to breakdown step **406** and analysis is continued until the termination condition is met.

Application of the Method of this Invention to a Specific Problem

[0088] The method is applied to finding data patterns and exceptions to both global and local rules or intelligences, analysis scope, based on any meaningful combination of members of the Performance Monitor dimension with respect to the Monitor Value measure via the construction of composite measures to accomplish a reduction in dataset dimensionality. The analysis can be further restricted by user specified thresholds and intra-measure and inter-measure conditions that constrain or restrict the analysis scope. Although such constraints on the analysis scope can be specified, in the present example, no thresholds or conditions are specified. Although this application of meta-exceptions and binning based on meta-exceptions is applied to a simple situation considering two composite measures, the method can be applied to n-composite measure problem.

[0089] For the purpose of controlled development, the goal of the application of the method is to analyze data associated with the performance monitor dimension and the monitor value measure and determine a global pattern, local patterns and exceptions and/or meta exceptions. Referring now to Figure 5, a cube having a dimensionality of four with one measure is shown represented by a partial data base schema showing the Performance Dimensions expanded to show the Computer dimension with its members Domain and Computer, the Configuration dimension with its member configuration, the Hour dimension with its members Hour and Min (minute) and the Performance Monitor dimension with its member Performance Monitor and the Measures dimension and its member Monitor Value.

[0090] Figure 6 depicts a crosstab of a composite measure through variable concatenation to reduce the dimensionality of the problem to be analyzed, where the problem has been reduced to finding an exceptional relation between Memory-Bytes Available and Memory-Pages per Sec based on the monitor value measure. In this cube, the Performance Monitors (Metrics) are arranged in a dimension, the tuple (Member of Performance Monitor Dimension, Member of Measures Dimension), which corresponds to step **404** of Figure 4, a wizard helps in the specification and customization of any user defined constructions.

[0091] Figure 7 shows an example of a wizard, which records user inputs related to a dimension level listing the measure metrics, where the user selections are formed into composite measures from the Performance Monitors dimension to analysis. In this figure, the user has selected "Memory-Bytes Available" and "Memory-Pages Per Sec" members from the Performance Monitor dimension.

[0092] Figure 8 shows the next screen of the wizard, which allows for the user to specify the measures that quantify the value of Performance Monitor members and to set constrains on the analysis scope. Thus, the composite measures may now be defined in MDX syntax as follows:

```
MEMBER [Performance Monitor].[Memory-Bytes Available_ Monitor
Value] AS '([Performance Monitor].[Memory-Bytes
Available],[Measures].[Monitor Value])'
MEMBER [Performance Monitor].[Memory-Pages per Sec_Monitor Value]
AS '([Performance Monitor].[Memory-Pages per
Sec],[Measures].[Monitor Value])'
```

[0093] The resulting composite measures are defined in the hierarchy of the "Performance Monitor" dimension so that the following query can be constructed in MDX code:

```
WITH
MEMBER [Performance Monitor].[Memory-Bytes Available_ Monitor
Value] AS '([Performance Monitor].[Memory-Bytes
Available],[Measures].[Monitor Value])'
MEMBER [Performance Monitor].[Memory-Pages per Sec_Monitor Value]
AS '([Performance Monitor].[Memory-Pages per
Sec],[Measures].[Monitor Value])'
SELECT
{[Performance Monitor].[Memory-Bytes Available_ Monitor Value],
[Performance Monitor].[Memory-Pages per Sec_Monitor Value] }
 ON COLUMNS
FROM Performance
```

[0094] The above MDX coding yields the results shown in Figure 9. In effect, for the defined analysis scope, the formation of the composite measure has reduced the cube dimensionality to form a virtual cube having the composite measures as dimensions as shown in Figure 10. The new composite measure includes members of the Performance Monitor dimension and serve as regular measure for further analysis. It should be noted that the creation of a composite measure is not necessary and, demonstrates that such customizations are easily and naturally allowed by the method. The composite measure concept is discussed more fully above in the "**On the Fly Binning**" section.

[0095] As shown in Figure 8, the user can specify constrains based on threshold values and conditions. Referring to Figure 11, the next wizard screen is shown, where further constrains related to the analysis search scope may be specified by limiting analysis across specified members, levels and dimensions.

[0096] For the current example, the selections in shown in Figure 11 indicate that every crosstab evaluated includes members from the Hour dimension at the Hour level. Members from other selected dimensions are included in the crosstabs as required by the iterative routines. Thus, the largest Non Empty search crosstab that can result from the above selections for studying the composite measure can be specified in MDX syntax as follows:

```
NONEMPTYCROSSJOIN(
      {[Computer].[Computer].MEMBERS},
      {[Configuration].[Configuration].MEMBERS},
      {[Hour].[Hour].MEMBERS}
      )
```

[0097] Having defined the composite measures and the analysis scope, the next step is to determine the "seed intelligence" for beginning the search for exceptions, local patterns and local pattern exceptions. For the current example, the analysis will focus on identifying exceptions. According to Step **408** of Figure 4, seed intelligence is determined in order to identify candidate crosstabs for analysis. The seed intelligence is constructed using the following MDX query:

```
/* 1) Composite Measure Definition */
WITH
MEMBER [Performance Monitor].[Memory-Bytes Available_ Monitor
Value] AS '([Performance Monitor].[Memory-Bytes
Available],[Measures].[Monitor Value])'
MEMBER [Performance Monitor].[Memory-Pages per Sec_Monitor Value]
AS '([Performance Monitor].[Memory-Pages per
Sec],[Measures].[Monitor Value])'
/* 2) Sample of High and Low valued members from Hour dimension
*/
SET [HourMembers] AS
'Union({TopPercent({[Hour].[Hour].members},10, [Performance
Monitor].[Memory-Bytes Available_ Monitor Value])},
{BottomPercent({[Hour].[Hour].members},10, [Performance
Monitor].[Memory-Bytes Available_ Monitor Value])})'
/* 3) Sample of High and Low valued members from Computer
dimension */
SET [ComputerMembers] AS
'Union({TopPercent({[Computer].[Computer].members}, 10,
[Performance Monitor].[Memory-Bytes Available_ Monitor Value])},
{BottomPercent({[Computer].[Computer].members},10, [Performance
Monitor].[Memory-Bytes Available_ Monitor Value])})'
/* 4) All members from Configuration dimension (Few Members) */
SET [ConfigurationMembers] AS
'{[Configuration].[Configuration].members}'
/* 5) Sample of High and Low valued tuples from Hour, Computer
and Configuration dimensional combinations */
Set [ScopeMembers] AS
'Union({TopPercent(NonEmptyCrossjoin({[Configuration].[Configur
ation].members},{[Hour].[Hour].members}
,{[Computer].[Computer].members}) , 10, [Performance
Monitor].[Memory-Bytes Available_ Monitor Value])},
{BottomPercent(NonEmptyCrossjoin({[Configuration].[Configuratio
n].members},{[Hour].[Hour].members}
,{[Computer].[Computer].members}),10, [Performance
```

```
Monitor].[Memory-Bytes Available_ Monitor Value])})'
/* 6) Correlation between the composite measures over Hour
dimension members */
MEMBER [Performance Monitor].[Correlation_Hours] AS
'CORRELATION({[HourMembers]}, [Performance Monitor].[Memory-Bytes
Available_ Monitor Value], [Performance Monitor].[Memory-Pages
per Sec_Monitor Value])'
/* 7) Correlation between the composite measures over Computer
dimension members */
MEMBER [Performance Monitor].[Correlation_Computer] AS
'CORRELATION({[ComputerMembers]}, [Performance Monitor].[Memory-
Bytes Available_ Monitor Value], [Performance Monitor].[Memory-
Pages per Sec_Monitor Value])'
/* 8) Correlation between the composite measures over
Configuration dimension members */
MEMBER [Performance Monitor].[Correlation_Configuration] AS
'CORRELATION({[ConfigurationMembers]}, [Performance
Monitor].[Memory-Bytes Available_ Monitor Value], [Performance
Monitor].[Memory-Pages per Sec_Monitor Value])'
/* 9) Correlation between the composite measures over all
dimensional tuples */
MEMBER [Performance Monitor].[Correlation_Overall] AS
'CORRELATION({[ScopeMembers]}, [Performance Monitor].[Memory-
Bytes Available_ Monitor Value], [Performance Monitor].[Memory-
Pages per Sec_Monitor Value])'
/* 10) Query to return all the correlation values */
Select {[Performance Monitor].[Correlation_Hours] ,[Performance
Monitor].[Correlation_Computer] , [Performance
Monitor].[Correlation_Configuration] , [Performance
Monitor].[Correlation_Overall] } ON COLUMNS
FROM Performance
```

[0098] The above MDX code yield the correlated results shown in Figure 12. In the MDX code, sections 2, 3, 4, and 5 relate to selecting the top 10% (or N% depending on sampling needs) and the bottom 10 % (or M% depending on sampling needs) of the members along the selected dimensions, based on one of the composite measures. Different techniques and bias (such as Prominent Members, User Defined Members, Random Selection, *etc.*) can be used to make this sampling more useful, while reducing the resources in deriving such preliminary intelligence.

[0099] Sections 6, 7, 8, and 9 relate to calculating correlations between the composite measures over the sampled members along the selected dimensions. It is important to note that Correlation is one of the many statistical techniques that can be employed here to understand the relationship between the composite measures in the analysis context.

[0100] Further, several permutations and combinations of members can be used to calculate behavior as discussed in step **410** of Figure 4. In that step, the method derives a useful "seed intelligence" which helps to guide the method towards generating and prioritizing data points or crosstabs for exception or pattern detection.

[0101] Referring again to Figure 12, the above determination of the seed intelligence indicates

that the dominant relationship between the composite measures is an inverse relationship between the members based on the measure. Although the method generated an inverse relationship between members, the user could have also defined an initial intelligence from his/her own knowledge of computers. Normally, paging occurs when memory availability is low and paging does not occur or occurs minimally when memory availability is high – an inverse relations. The method is designed to detect situations (Dimensional Contexts) in which this general relationship between available memory and paging is not true. The goal of the method is to identify such situations quickly on a user demand basis, a periodic basis or a continuous basis. Thus, exceptions should occur where Memory Availability and Memory Paging are positively or directly correlated.

[0102] Referring now to Figure 13, a plot **500** of the seed intelligence of composite measure values across the entire selected analysis scope is shown, where the seed intelligence is shown a straight line **502** having a negative slope derived from the analysis shown in Figure 12. The plot **500** has been divided into a grid **504** having nine regions **506** corresponding to low, medium and high values for the two correlated members, Memory Availability on the vertical axis and Memory Paging on the horizontal axis. Because the seed intelligence is an inverse relationships, regions of exceptions **508a&b** can be easily identified. Region **508a** corresponds to data points that have high values for both members simultaneously, a clear violation of the seed intelligence rule; while region **508b** corresponds to data points that have low values for both members simultaneously, another type of clear violation of the seed intelligence. Although the seed intelligence generated in the example is a simple straight line, the same type of candidate regions identification formulation can be used for even very complex curve fit correlations, because the regions where exceptions would be found are as well defined after seed intelligence construction as the seed intelligence is itself.

[0103] Now that the two shaded regions **508a** and **508b** have been identified, which again represent regions where existence of any data points are potentially contrary to the seed intelligence, the method prioritizes the analysis across the regions to converge on the anomalies quickly. Moreover, conducting analysis by regions or by even smaller binned subregions within the main exception regions, helps to break down a large problem into smaller units; thereby improving scalability.

[0104] Referring now to Figure 14, the concept of binned composite members or the construction of virtual data points is shown as discussed in step **412** of Figure 4. In Figure 14,

the seed intelligence **502**, the grid **504**, the nine regions **506** and the exception regions **508a&b** are shown. Because the grid **504** is a very course binning construct, the use of these course bins **506** is not a preferred method for developing local intelligence within the two exception regions **508a&b** – a single bin **506** cover almost the entire exception regions **508a&b**. Thus, the preferred method to develop local intelligence uses a finer binning procedure. In Figure 14, the region **508a** is analyzed using bins that are much smaller than the grid bins **506**, where a horizontal bar **510** shows the binning along the Memory Availability axis and the vertical bar **512** shows the binning along the Memory Paging axis. The method then determines whether any data falls within an intersection **514** of the two bars **510** and **512**. If data is found within the intersection **514** and in adjacent regions as the analysis progresses, then a local intelligence can be constructed as represented by a line segment **516**. This same process can be applied to the exception region **508b**, which may yield a different local intelligence represented by a line segement **518**.

[0105] This methodology can be illustrated by the following MDX query:

```
WITH
/* 1) Bin along Memory Paging Monitor Value Composite Measure */
MEMBER [Performance Monitor].[Memory-Pages per Sec_(0 To 5)] AS
'([Performance Monitor].[Memory-Pages per
Sec],[Measures].[Monitor Value])'
/* 2) Bin along Memory Availability Monitor Value Composite
Measure -Scaled Down*/
MEMBER [Performance Monitor].[Memory-Bytes Available_(30 To 60)]
AS '([Performance Monitor].[Memory-Bytes
Available],[Measures].[Monitor Value])/1048576'
SET [BININTERSECTION] AS
'{NONEMPTYCROSSJOIN({[Computer].[Computer].MEMBERS},{
[Configuration].[Configuration].MEMBERS},{[Hour].[Hour].MEMBERS
})}'
SELECT
FILTER( { [BININTERSECTION]}, (( [Performance Monitor].[Memory-
Pages per Sec_(0 To 5)] >=0 AND [Performance Monitor].[Memory-
Pages per Sec_(0 To 5)] <=5) AND ([Performance Monitor].[Memory-
Bytes Available_(30 To 60)] >=30 AND [Performance
Monitor].[Memory-Bytes Available_(30 To 60)] <=60))) ON COLUMNS,
{[Performance Monitor].[Memory-Pages per Sec_(0 To
5)],[Performance Monitor].[Memory-Bytes Available_(30 To 60)]} ON
ROWS
FROM Performance
```

[0106] "The MDX code yield the results shown in Figure 15, which shows the crosstab definition being presented in terms of dimensional combination and prevailing conditions and thresholds of composite members. This definition protocol aids in the understanding of results when exceptions and/or patterns are detected and presented to the end-user. An example of a crosstab definition in case of pattern detection or hybrid analysis could be crosstab composed

of members of affinity group and affinity thresholds such as support, confidence, improvement *etc.*

[0107] In the query, "[BININTERSECTION]" is defined by crossjoining all dimensional members in the analysis scope. As the analysis progresses, various crosstabs defined by the binned values of composite measures along the analysis scope are evaluated. The dimensions and composite members can be arranged in several permutations, such that more meaningful crosstabs are generated for analysis based on user and algorithm intelligence. While dimensional permutations in violation of expected behavior are primary exceptions, the same analysis iteration may also be used for analysis using other statistical techniques which may only utilize the dimensional context for analysis to detect secondary exceptions or patterns.

[0108] Referring now to Figure 16, an iteration of analysis is shown where global ("seed intelligence"), regional and local behavior of composite measure is determined and revised. Here the size of the crosstab being subjected to analysis is governed by the size of the composite measure(s) bin range. Regardless of the composite measure bin sizes, the aggregated system has the same workload to evaluate the crosstab qualifying the prevailing thresholds and conditions.

[0109] In small to medium aggregated structures, this may not be of a significant overhead; however for larger structures analysis routines can be structured such that the "data slice" qualification overhead is split into smaller units. For example, consider the analysis routines below:

[0110] 1. Pivot on composite measure(s) and split, "data slice", qualification overhead as described conceptually in the following pseudo MDX code:

```
/* Composite Measure(s) iteration */
For composite measures iteration ((CompositeA Bin (n), CompositeB
Bin (m)…), Bin Permutation (u))
{
      /* "data slice" to be qualified against prevailing
threshold */
      For dimensional iteration (Level (Diml (i), Dim2 (j), Dim3
(k) …), Permutation (l))
      {
            /* "data slice" to be qualified being split into
            smaller units*/
            For member set iteration (Set (Diml (o), Dim2 (p),
            Dim3 (q)…))
            {
                  /* The qualified spit unit of "data slice" is
                  analyzed */
                  Exception/Pattern detection iteration
                  (Statistical Technique (t))
            }
      }
}
```

[0111] 2. Pivot on dimensional iteration(s) and **do not** split, "data slice", qualification overhead as described conceptually in the following pseudo MDX code:

```
/* "data slice" to be qualified against prevailing threshold */
For dimensional iteration (Level Members (Dim1 (i), Dim2 (j),
Dim3 (k) ...), Permutation (1))
{
      /* Composite Measure(s) iteration */
      For composite measures iteration ((CompositeA Bin (n),
      CompositeB Bin (m)...), Bin Permutation (u))
      {
            /* The qualified spit unit of "data slice" is
            analyzed */
      Exception/Pattern detection iteration (Statistical Technique (t))
      }
}
```

[0112] The Iteration process involves all possible permutations of member sets and composite measure bins. The composite measure bins can be substituted by cluster group list or affinity groups list etc. in case of pattern or hybrid analysis inline with example "a" as set forth in the following MDX code:

```
WITH

/* 1) Bin along Memory Paging Monitor Value Composite Measure */
MEMBER [Performance Monitor].[Memory-Pages per Sec_(0 To 5)] AS
'([Performance Monitor].[Memory-Pages per
Sec],[Measures].[Monitor Value])'

/* 2) Bin along Memory Availability Monitor Value Composite
Measure -Scaled Down*/
MEMBER [Performance Monitor].[Memory-Bytes Available_(30 To 60)]
AS '([Performance Monitor].[Memory-Bytes
Available],[Measures].[Monitor Value])/1048576'

/* 3) Current iteration Split Analysis Scope*/
SET [BININTERSECTION] AS
'{NONEMPTYCROSSJOIN(Subset({[Computer].[Computer].MEMBERS},100,
50),SubSet({[Hour].[Hour].Members}, 0,50),
Subset({[Configuration].[Configuration].Members}, 0, 50))}'

/* 4) Query to return the qualified data slice*/
SELECT
FILTER( { [BININTERSECTION]}, (( [Performance Monitor].[Memory-
Pages per Sec_(0 To 5)] >=0 AND [Performance Monitor].[Memory-
Pages per Sec_(0 To 5)] <=5) AND ([Performance Monitor].[Memory-
Bytes Available_(30 To 60)] >=30 AND [Performance
Monitor].[Memory-Bytes Available_(30 To 60)] <=60))) ON COLUMNS,
{[Performance Monitor].[Memory-Pages per Sec_(0 To
5)],[Performance Monitor].[Memory-Bytes Available_(30 To 60)]} ON
ROWS

FROM Performance
```

[0113] The MDX code yields the results shown in the crosstab of Figure 17. Section 3 of the above MDX shows that we are including 50 Members (variable) at a time from each dimension;

this breaks down the overhead associated with qualifying the data slice across multiple iterations for the purposes of scalability and fast convergence. The number of members included from each dimension is influenced by the dimension size and the number of dimensions in the cube. Members may be selected sequentially, randomly, based on ranking, inter and intra dimensional affinities and/or based on non -sparse member and/or tuple list persisted through earlier iterations and analysis runs. The main goal being to converge on exceptional data slices as efficiently as possible.

[0114] In section 4, we use the binned threshold values of composite measures to qualify the data slice; these thresholds may be determined automatically by the algorithm and/or specified by the user, within the continuum of upper and lower values of these composite measures. The members per dimension and bin range thresholds are also affected by the data characteristics. The size of the analysis data slice being evaluated needs to optimal for statistical technique being employed to identify exceptions and patterns and minimize "false positive" exceptions or patterns.

[0115] The evaluation routines illustrated in example "a" and "b" above, allow for effective use of parallel processing. Further, when an anomaly/pattern is detected, separate independent processes may be launched to further qualify the finding by including other dimensional and measure entities, *etc*.

[0116] In the preceding discussion related to the analysis crosstab creation and analysis guidance, it is important to note that the algorithmic routines eventually evaluate entire analysis space specified by the user. However, the algorithm prioritizes the evaluation across smaller units of the analysis, based on algorithmic and/or user biases, with the goal of converging on exception or patterns faster. Exceptions and patterns may be determined by using combination of statistics and data mining techniques, such as exponential trend, Chi-Squared Deviancies, ANOVA, MANOVA, Cramer's coefficient, entropy, Classical Clustering, SOM, Affinity Analysis etc., depending on the objective of the analysis and the analysis scope definition.

[0117] As the analysis of various crosstabs proceeds, the normal expected behavior is constantly updated and Local, Regional and Global versions of such intelligence are maintained and utilized for generating high exception probability crosstabs and prioritizing the analysis across these crosstabs as has been discussed earlier. In some cases the crosstab definitions along with the prevailing thresholds may be persisted so that it can be reutilized in subsequent analysis runs. An alternative application of the concept of Composite Measures and Binned Ranges is in Ad-

Hoc reporting, which is also utilized for presenting the exceptions and patterns in the If ... Then... form.

[0118] Use of composite measures in the creation of candidate crosstabs helps in overcoming the hard boundaries posed by pre-aggregated structures and offers new data points for analysis. Seed (or Global), regional and local behavioral intelligence of composite measures and user defined biases can be effectively used to guide analysis to converge on anomalies and patterns quickly and yield better insights. Partitioning the analysis scope into smaller units helps with scalability and effective analysis processing. Composite measure definitions and bin ranges thresholds when presented with the exception and patterns helps with ease of understanding.

[0119] During iteration, when an anomalous cellset is detected, a new process (in parallel or for post–processing) can be started to further qualify the anomaly and investigate behavior along previously unconsidered conditions. The process interface will optionally allow for capturing user biases related to prioritization of such new investigative processes and also to define a new scope along previously unconsidered conditions.

[0120] This technique provides a guided path for the statistical techniques to find exceptions and converge on the anomaly fast. However, it is important to note that the ultimate search space is defined by the multi-dimensional database and optionally by user selection. The algorithm evaluates the entire search space eventually; however, process intelligence and/or user biases define priorities of such evaluation. Thus, leveraging the synergy between automated algorithms and the subject matter expert, as represented by the user. The statistical outlook (for the entire population) on a sample data (for performance reasons) is revised intelligently (in local zones) to minimize false exceptions. The results of such ans analysis could be the identification of local trends within data bins as shown in Figure 18.

[0121] This processing can be done in incremental passes – where new "Normal" relationship is defined based on detected exceptions, which now serve as new sample population.

[0122] For example, crosstab combinations are computed with the goal of converging to potential anomalies first, *e.g.*, along Computers with High Paging and High Memory Availability, so that most prominent Dimensional Members are analyzed First. Such prioritization of binned data is shown in Figure 19, where the data is analysis in the order shown. Thus, the problem definition is divided along multiple parameters with the goal to converge to anomalies fast. While Bin Combinations define scoping based on Crosstab Qualifications, simultaneous use of prominent members (along dimensions/levels *etc.*, *e.g.*, computers with

higher memory availability, further scopes the problem, such that prominent anomalies are detected fast, a result of such an analysis is shown in Figure 20, where the anomaly is shown in the hatched circle.

[0123] Alternately, combinations of statistical tools can be utilized do detect variations/anomalies: for example, exponential trend, Chi-Squared Deviancies, Cramer's coefficient, entropy, Classical Clustering, SOM *etc.*

[0124] Other statistical techniques that can be used with the present methodology including Tendency Analysis, where the main idea is to be able to a collection of "profile" curves representing tendencies in the data. To "detect" significant changes for each event (combination of dimension members), it is enough to detect changes in curve in the tendency plots.

[0125] As a summary, the user can select the dimensions he/she wants to use as an event for the analysis. The algorithm generates several queries to get the tendency curve and a simple technique will be used to detect significant changes. If those changes are significant, the algorithm reports this combination of members to the user. This "Tendency Analysis" can provide information only about when a deviation in the tendency occurs.

[0126] Besides Tendency Analysis, condition analysis can be used. If the user wants to find out which conditions are tending to fluctuate, then the method will need to detect all combinations that move their tendency one way or the other. Thus, condition analysis provides information about where a change in the tendency occurs. The method can ask the user to select if he/she wants a positive or negative tendency and we can show all conditions that have the requested tendency.

[0127] One important aspect of the methods of this invention relates to determining bin sizes, and what approach to take to iterate across Bins, *e.g.*, independent bins (across each composite measure) or process bins one at a time as shown in Figures 21A&B.

[0128] The one bin at a time approach (as opposed to considering two Bins - along two composite Measure simultaneously) may minimize the effect of Bad Bins and provide sufficient data points to identify exceptions. Most importantly, Binning allows establishing a context of crosstab definition, which when subjected to statistical evaluation, would yield results in a context that can be readily and easily understood.

[0129] The method also compares seed intelligence on each cycle and displays a screen as shown in Figure 22, if the seed intelligence is confirmed.

[0130] Any patents or publications mentioned in this specification are indicative of the levels

of those skilled in the art to which the invention pertains. These patents and publications are herein incorporated by reference to the same extent as if each individual publication was indicated to be incorporated by reference specifically and individually.

[0131] One skilled in the art will readily appreciate that the present invention is well adapted to carry out the objects and obtain the ends and advantages mentioned, as well as those inherent therein. It will be apparent to those skilled in the art th at various modifications and variations can be made in practicing the present invention without departing from the spirit or scope of the invention. Changes therein and other uses will occur to those skilled in the art which are encompassed within the spirit of the invention as defined by the scope of the claims.

CLAIMS

We claim:

1. A method implemented on a computer to alter a dimensionality of a multi-dimensional database hierarchical structure, iteratively and dynamically, comprising the steps of:

providing a multidimensional database having a native schema,

selecting a plurality of members and at least one measure from the schema,

merging at least one of the plurality of members and the at least one measure to form an imaginary schema,

where the imaginary schema enhances, increases and/or makes more efficient data processing of the data in the dataset so that qualified data points are made available to various data mining algorithms and where the imaginary schema alters a dimensionality of the database.

2. The method of claim 1, wherein the dimensionality is increased or reduced or the dimensionality of any part of the database is increases or decreased.

3. A method for detecting data anomalies, exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically and virtually generating data points based on base aggregated data including the steps of selecting at least one multi-dimensional dataset, preferably in the form of an OLAP cube, and at least one measure associated with the data dimension in the dataset; capturing a scope of analysis and constraints from a user; constructing a virtual database schema from a native database schema of the dataset to reduce or expand the dimensionality of the dataset as a whole or in regions of interest, while maintaining the associated measure or producing a composite measure from more than one measure; selecting a limited number of data values from the entire dataset or the part of interest; creating an initial global rule, "seed intelligence" describing the behavior of the measure with respect to the selected, limited number of data values; determining data regions that would violate the initial global rule; prioritizing the regions; searching the dataset for data the satisfies the initial seed intelligence and that falls within the regions forming regional datasets; and reporting the regional datasets.

4. Amethod for detecting data anomalies, exceptions or meta exceptions and/or identifying

patterns in base aggregated data and/or automatically and virtually generating data points based on base aggregated data including the steps of selecting at least one multi-dimensional dataset, preferably in the form of an OLAP cube, and at least one measure associated with the data dimension in the dataset; capturing a scope of analysis and constraints from a user; constructing a virtual database schema from a native database schema of the dataset to reduce or expand the dimensionality of the dataset as a whole or in regions of interest, while maintaining the associated measure or producing a composite measure from more than one measure; selecting a limited number of data values from the entire dataset or the part of interest; creating an initial global rule, "seed intelligence" describing the behavior of the measure with respect to the selected, limited number of data values; determining data regions that would violate the initial global rule; prioritizing the regions; searching the dataset for data the satisfies the initial seed intelligence forming a compliance dataset and that falls within the regions forming regional exception datasets; if the regional exception datasets are not null (empty) or do not contain too few data points to support statistical analysis, creating regional intelligence or local intelligence; determining datapoints within each regional exception dataset that represent exceptions to the local intelligence; and reporting the results.

5. A method for detecting data anomalies, exceptions or meta exceptions and/or identifying patterns in base aggregated data and/or automatically and virtually generating data points based on base aggregated data including the steps of selecting at least one multi-dimensional dataset, preferably in the form of an OLAP cube, and at least one measure associated with the data dimension in the dataset; capturing a scope of analysis and constraints from a user; constructing a virtual database schema from a native database schema of the dataset to reduce or expand the dimensionality of the dataset as a whole or in regions of interest, while maintaining the associated measure or producing a composite measure from more than one measure; selecting a limited number of data values from the entire dataset or the part of interest; creating an initial global rule, "seed intelligence" describing the behavior of the measure with respect to the selected, limited number of data values; determining data regions that would violate the initial global rule; prioritizing the regions; searching the dataset for data the satisfies the initial seed intelligence forming a compliance dataset and that falls within the regions forming regional exception datasets; if the regional exception datasets are not null (empty) or do not contain too few data points to support statistical analysis, creating regional intelligence or local intelligence;

determining datapoints within each regional exception dataset that represent exceptions to the local intelligence; update the initial seed intelligence with the local intelligences properly weighted to form an updated seed intelligence; comparing the updated seed intelligence; if the updated seed intelligence is significantly different from the initial seed intelligence, replacing the initial seed intelligence with the updated seed intelligence; repeating the previous three steps, until there is no significant change between the seed intelligence from the previous iteration and this iteration; and reporting the results.

6. The method of claim 5, further comprising the step of determining whether a termination condition has been met, where failure to met the condition would restart the analysis construction of the scope of analysis step and the method steps would be continued until the condition is met.

7. A method for finding global and local intelligences quickly including the steps of: capturing an analysis scope, a breakdowning the analysis scope into one or more logical units or combinations thereof, optionally specifying constraints on the analysis scope; establishing a seed intelligence from a sample data population, from user input or a combination of data sampling and user input, identifying data regions that represent exceptions to the seed intelligence, establishing local intelligence in each exception region, if non empty, updating seed intelligence with local intelligences or forming a composite intelligence of an updated seed intelligence and local intelligences; testing to determine if the seed intelligence or composite intelligence from the last cycle is significantly different than the seed intelligence or composite intelligence of this cycle, exiting changes are insignificant or returning to the identifying step if significant changes occurred for iteration until convergence is achieved, where after convergence, the method will have constructed a consistent intelligence, seed or composite, for describing the data behavior and will have identified exceptional regions, local intelligence associated with the regions and exceptions to the local rules..

8. The method of claim 7, further comprising testing the intelligences to determine if a termination condition has been met.

9. A method for constructing an intelligence models including an overall or global intelligence and local intelligences using the methods of claims 3-8, which generates the

intelligences from the analysis of data in multidimensional databases, relational or OLAP, and in the use the intelligence model to predict further data behavior.

10. A method for constructing libraries of intelligence models, each model including an overall or global intelligence and local intelligences using the methods of claims 3-8, which generates the intelligences from the analysis of data in multidimensional databases, relational or OLAP.

11. A method for using the library of intelligence models to classify data behavior and as a tool for predicting the behavior of classified data and in the use the intelligence models to predict further data behavior, where the models are generated by the method of claims 3-8.

12. A computer having stored thereon code sufficient to implement the method of any of the claims 1-11.

13. A computer readable medium having stored thereon code sufficient to implement the method of any of the claims 1-11.

14. A system for finding global and local data patterns and exceptions to both the global pattern and the local pattern, comprising:

an analysis scope capture and definition module,

a breakdown module for breaking the analysis scope into logical units or combinations of logical units,

a seed intelligence module that determines a seed intelligence (global rule) from a limited data selection from the data to be analyzed;

a determine exception candidate region module where regions of data which would violate the seed intelligence are identified, prioritized and analyzed inline with the analysis of the seed intelligence guess,

a determine local intelligence and identify local intelligence exceptions and compare the local intelligence to the seed intelligence,

a create an updated seed intelligence module, where the updated seed intelligence and test the updated seed intelligence against the current seed intelligence and repeat the analysis

until the updated seed intelligence and current seed intelligence differ by only an insignificant amount.

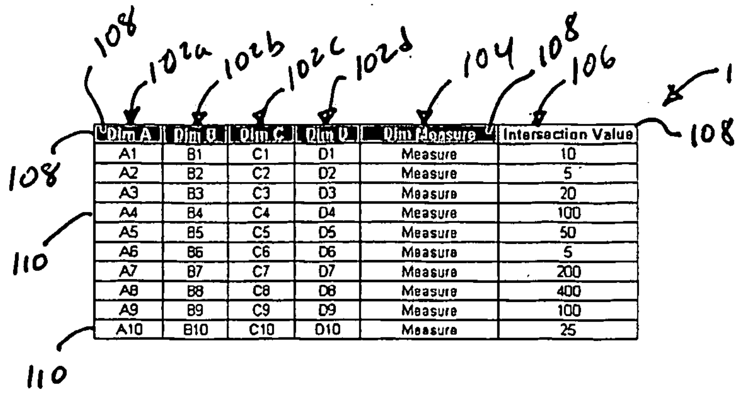
15. A computer having stored thereon code sufficient to implement the system of claim 14.

16. A computer readable medium having stored thereon code sufficient to implement the system of claim 14.

17. An analysis wizard including a sequence of windows designed to define an analysis scope, define meta dimensions for construction of imaginary database schema, and to defined user customizations or constraints of the imaginary database schema.

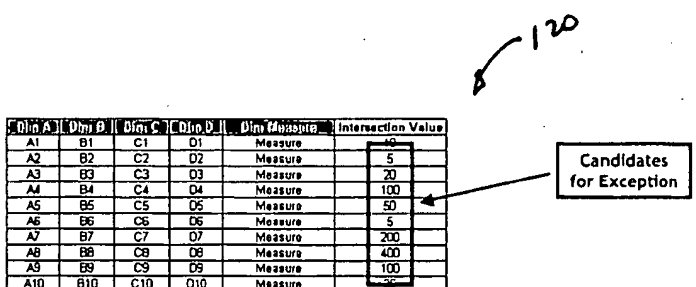
18. A computer having stored thereon code sufficient to implement the wizard of claim 17.

19. A computer readable medium having stored thereon code sufficient to implement the wizard of claim 17.

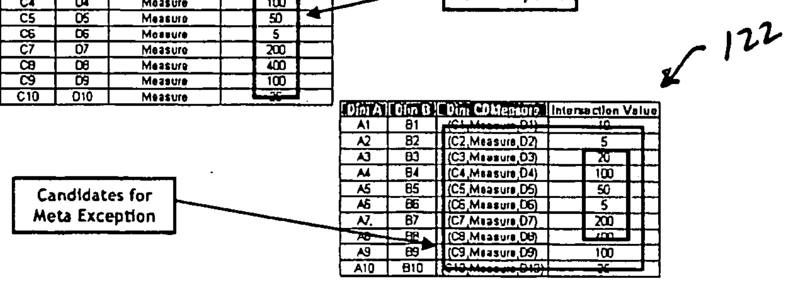


Dim A	Dim B	Dim C	Dim D	Dim Measure	Intersection Value
A1	B1	C1	D1	Measure	10
A2	B2	C2	D2	Measure	5
A3	B3	C3	D3	Measure	20
A4	B4	C4	D4	Measure	100
A5	B5	C5	D5	Measure	50
A6	B6	C6	D6	Measure	5
A7	B7	C7	D7	Measure	200
A8	B8	C8	D8	Measure	400
A9	B9	C9	D9	Measure	100
A10	B10	C10	D10	Measure	25

FIG. 1A



Dim A	Dim B	Dim C	Dim D	Dim Measure	Intersection Value
A1	B1	C1	D1	Measure	10
A2	B2	C2	D2	Measure	5
A3	B3	C3	D3	Measure	20
A4	B4	C4	D4	Measure	100
A5	B5	C5	D5	Measure	50
A6	B6	C6	D6	Measure	5
A7	B7	C7	D7	Measure	200
A8	B8	C8	D8	Measure	400
A9	B9	C9	D9	Measure	100
A10	B10	C10	D10	Measure	25



Dim A	Dim B	Dim CDMeasure	Intersection Value
A1	B1	(C1,Measure,D1)	10
A2	B2	(C2,Measure,D2)	5
A3	B3	(C3,Measure,D3)	20
A4	B4	(C4,Measure,D4)	100
A5	B5	(C5,Measure,D5)	50
A6	B6	(C6,Measure,D6)	5
A7	B7	(C7,Measure,D7)	200
A8	B8	(C8,Measure,D8)	400
A9	B9	(C9,Measure,D9)	100
A10	B10	(C10,Measure,D10)	25

FIG. 1B FIG. 1C

Meta Exception Definition (1)

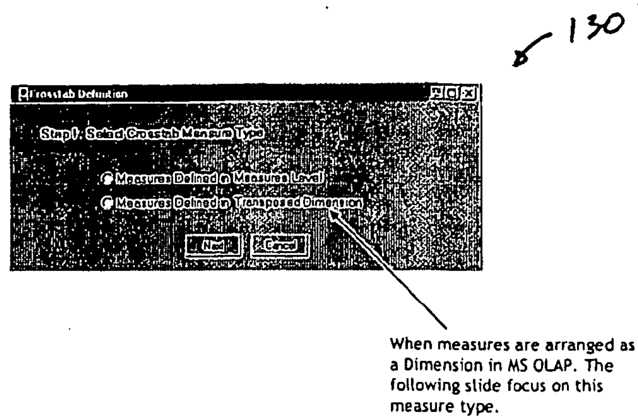


FIG. 2A

Meta Exception Definition (2)

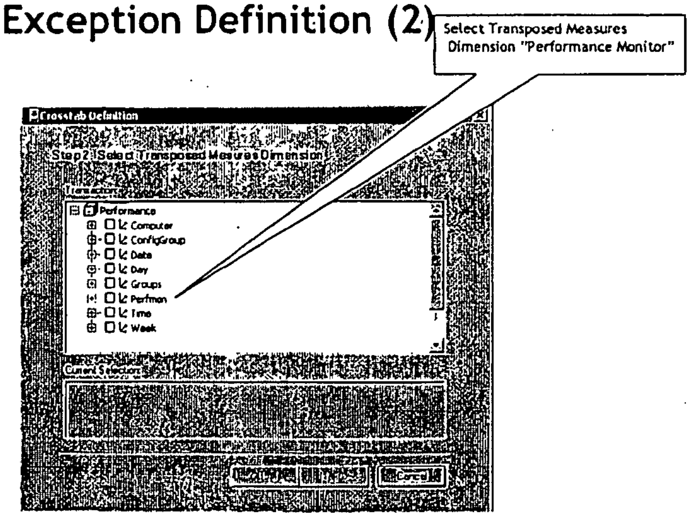


FIG. 2B

Meta Exception Definition (3)

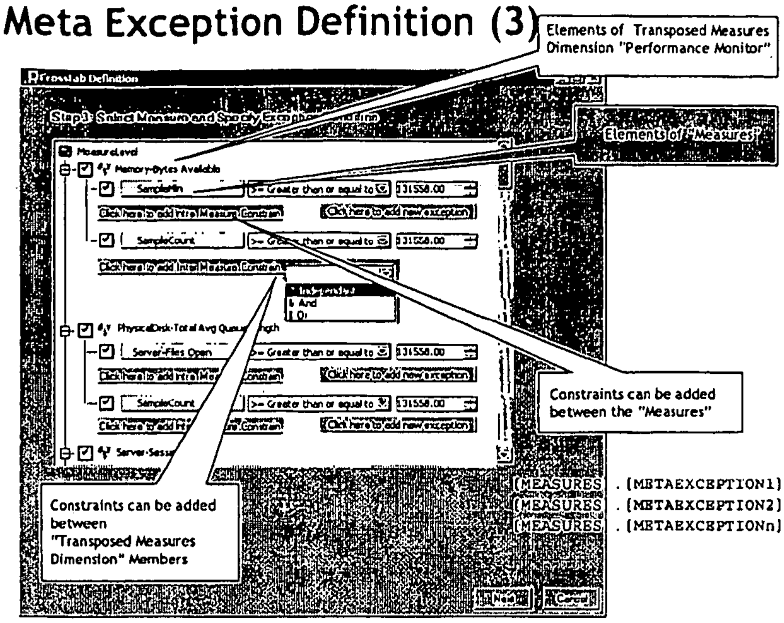


FIG. 2C

Crosstab Definition (4)

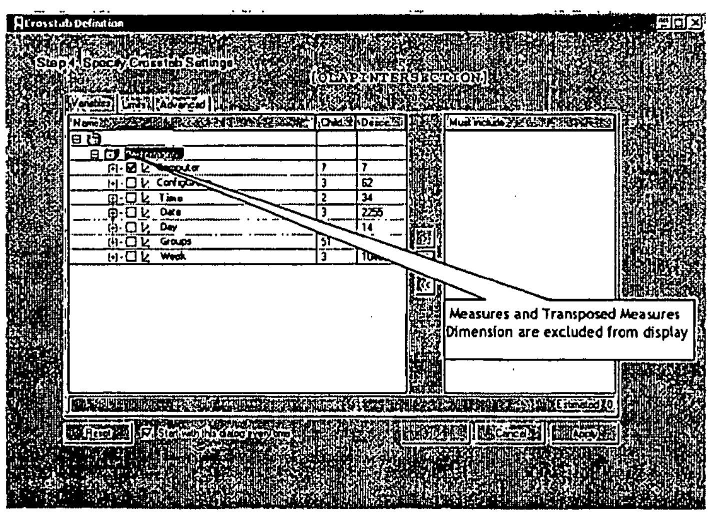


FIG. 2D

	CustomersUnitSales_BIN3	CustomersSalesAverage_BIN4
ProductUnitSales_BIN1	1,689.00	353.00
ProductUnitSales_BIN2	364.00	1,926.00

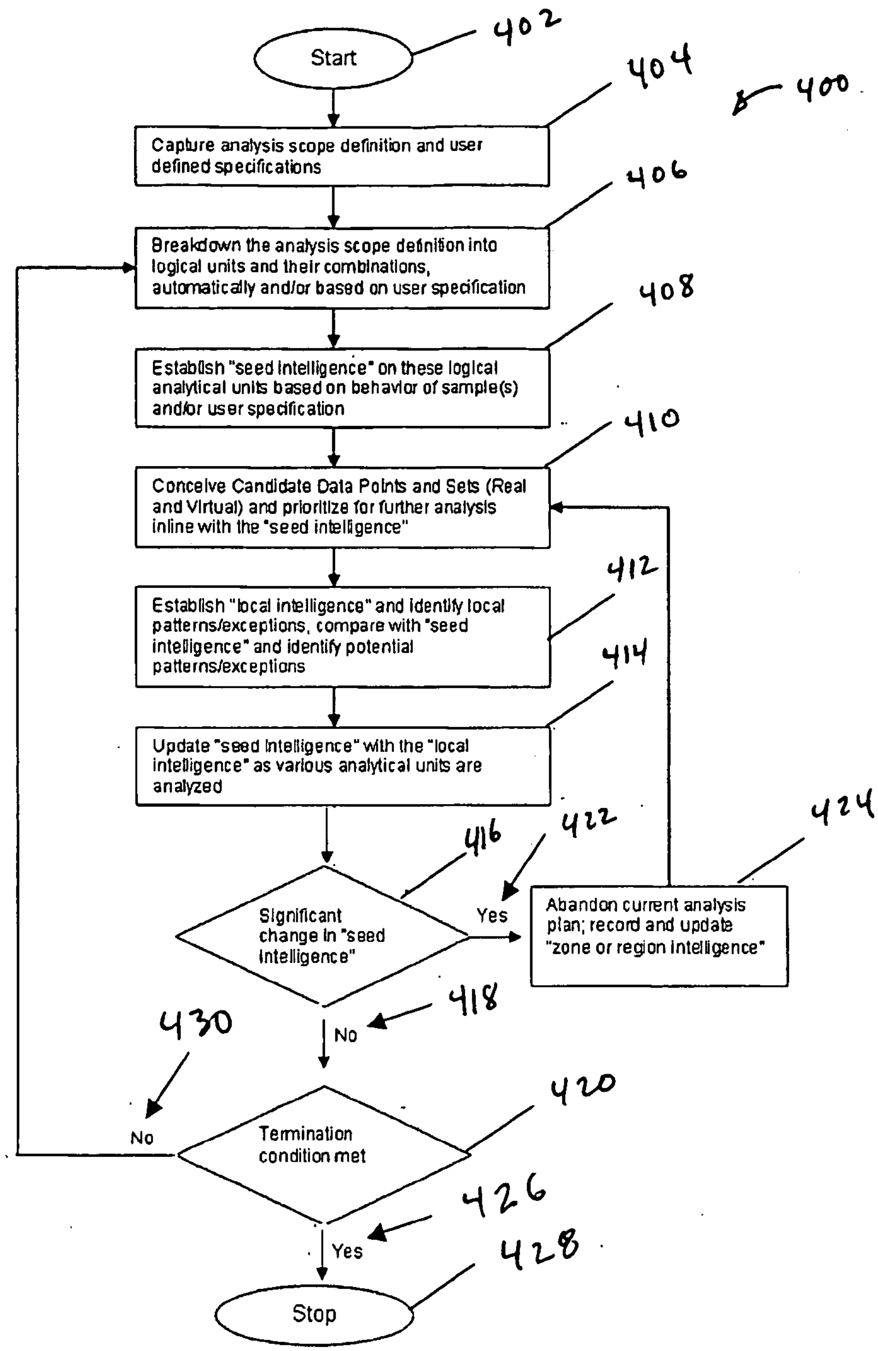
FIG. 3A

	CustomersEducationUnitSales_BIN3	CustomersEducationStoreCost_BIN4
ProductStoreTypeUnitSales_BIN1	6,336.00	3,176.00
ProductStoreTypeSalesAverage_BIN2	41,265.00	$\boxed{203,775.00}$

FIG. 3B

		Promotion_BIN1	Promotion_BIN2
Customers_BIN1	Product_BIN1	$\boxed{502.00}$	1,548.00
	Product_BIN2		
Customers_BIN2	Product_BIN1	151.00	280.00
	Product_BIN2		

FIG. 3C



402
Start
404
400
Capture analysis scope definition and user defined specifications
406
Breakdown the analysis scope definition into logical units and their combinations, automatically and/or based on user specification
408
Establish "seed intelligence" on these logical analytical units based on behavior of sample(s) and/or user specification
410
Conceive Candidate Data Points and Sets (Real and Virtual) and prioritize for further analysis inline with the "seed intelligence"
412
Establish "local intelligence" and identify local patterns/exceptions, compare with "seed intelligence" and identify potential patterns/exceptions
414
Update "seed intelligence" with the "local intelligence" as various analytical units are analyzed
416
422
424
Significant change in "seed intelligence"
Yes
Abandon current analysis plan; record and update "zone or region intelligence"
418
No
430
No
420
Termination condition met
426
Yes
428
Stop

FIG. 4

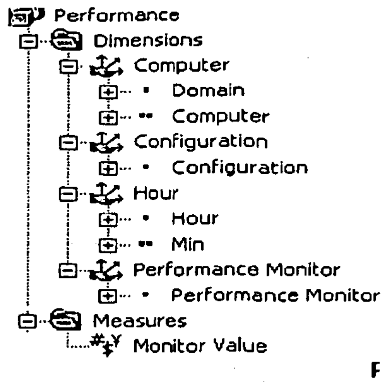


FIG. 5

	JJMemory-Bytes Available	JJMemory-Pages per Sec
	JJMonitor Value	JJMonitor Value
JJTOTALS	11,892,278,988,800.00	69,128.77

FIG. 6

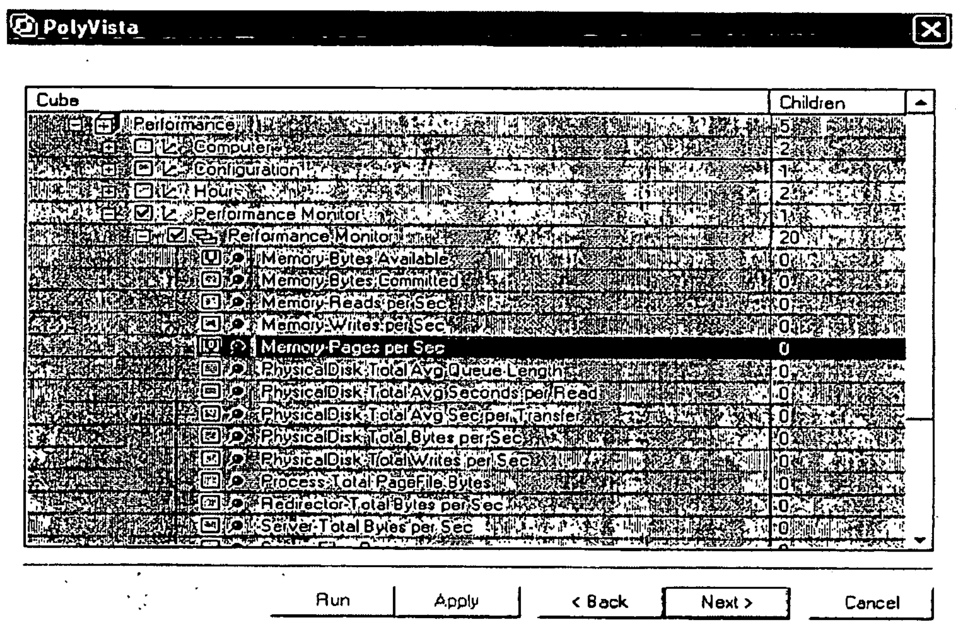


FIG. 7

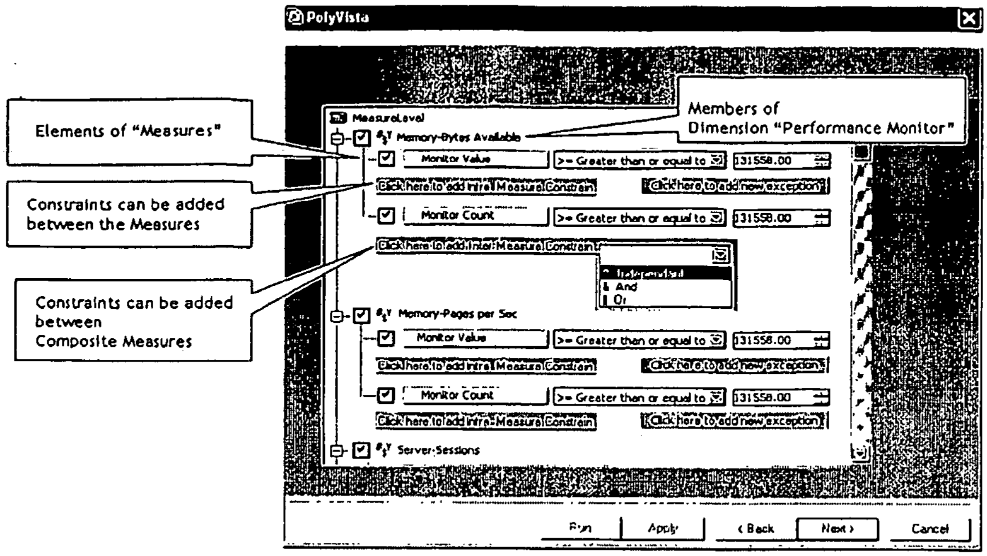


FIG. 8

Memory-Bytes Available_ Monitor Value	Memory-Pages per Sec_Monitor Value
11,892,278,988,800.00	69,128.77

FIG. 9

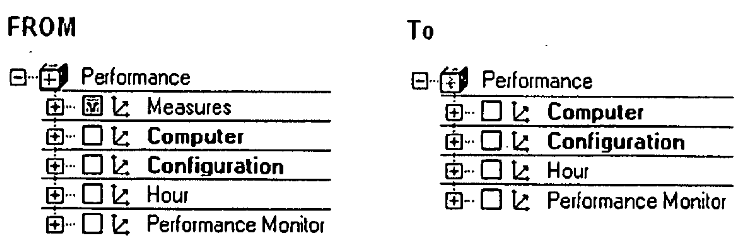


FIG. 10

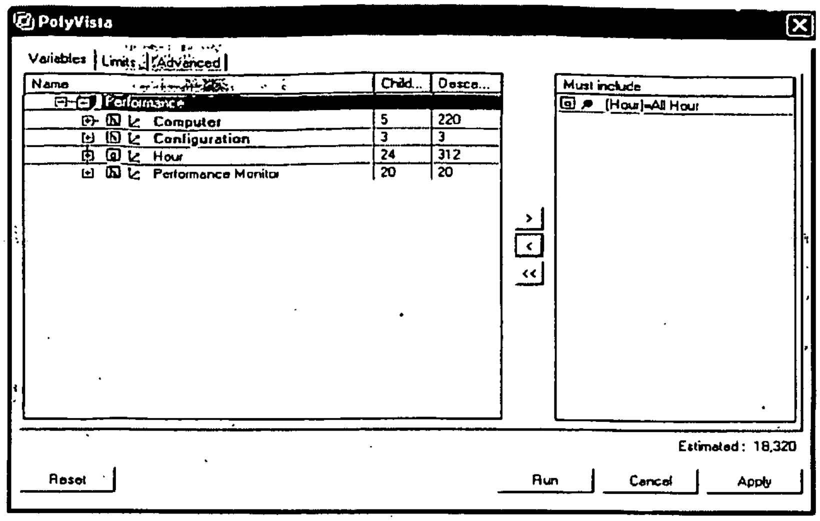


FIG. 11

Correlation_Hours	Correlation_Computer	Correlation_Configuration	Correlation_Overall
-0.92	-0.32	1.00	-0.26

FIG. 12

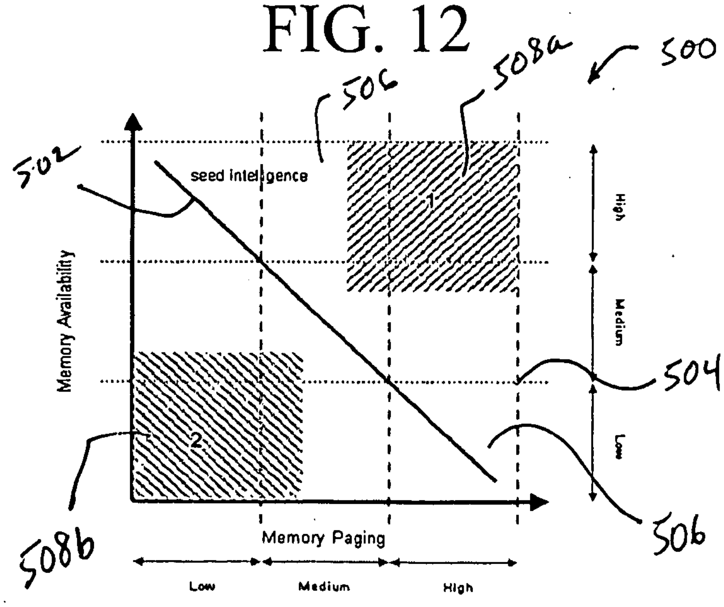


FIG. 13

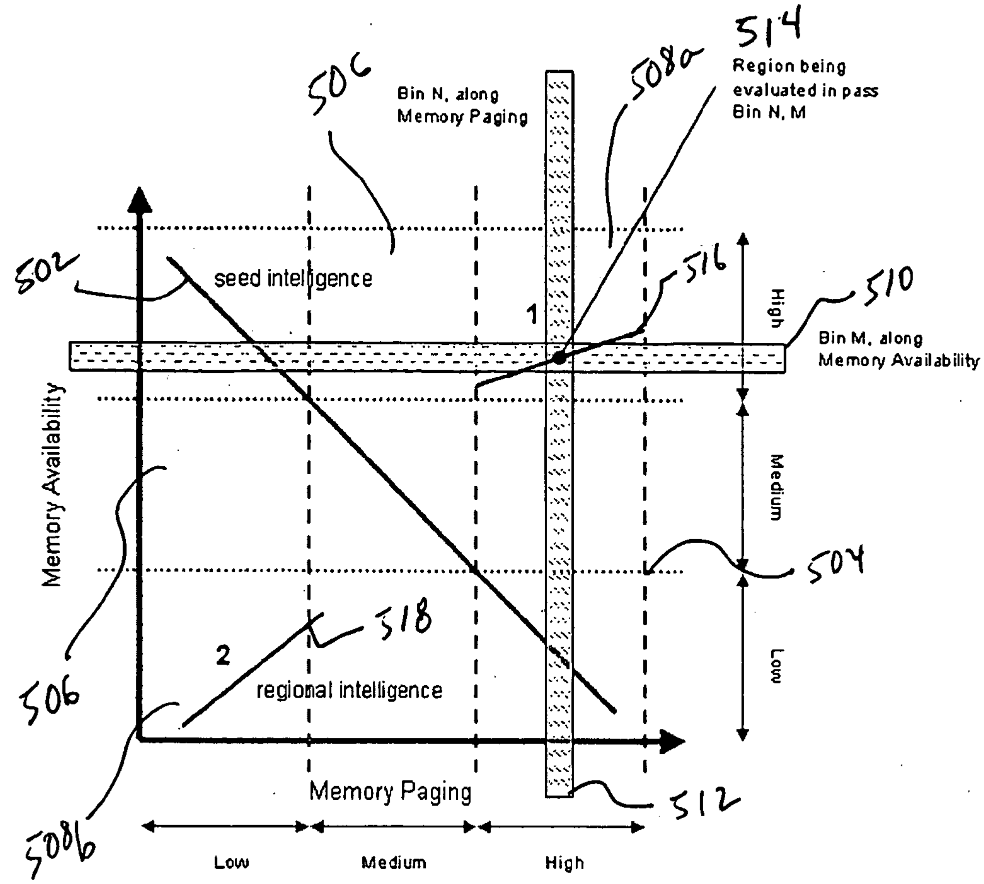


FIG. 14

			Memory-Pages per Sec_(0 To 5)	Memory-Bytes Available_(30 To 60)
Computer176	CG1	18		47.61
		19		45.79
		20		42.37
		22		58.62
		23		50.86
Computer198	CG1	17		53.82
		18		56.93
Computer211	CG3	6	4.19	58.27
Computer249	CG3	23		55.68
Computer018	CG2	10		56.79
		13		58.63
Computer032	CG2	23		58.68
		0		55.90
		1		57.24

FIG. 15

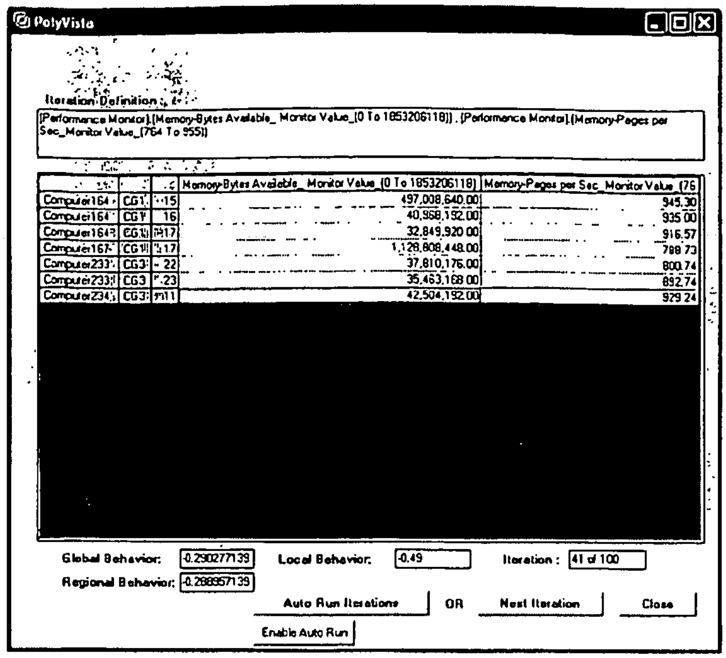


FIG. 16

			Memory-Pages per Sec_(0 To 5)	Memory-Bytes Available_(30 To 60)
Computer198	17	CG1		53.82
	18	CG1		56.93
Computer211	6	CG3	4.19	58.27

FIG. 17

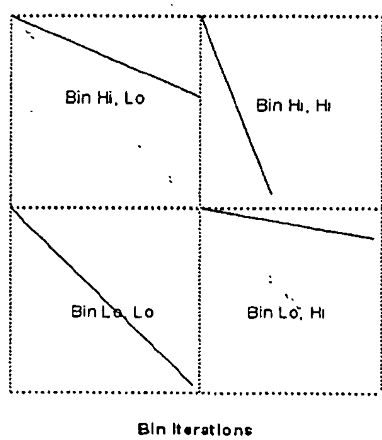


FIG. 18

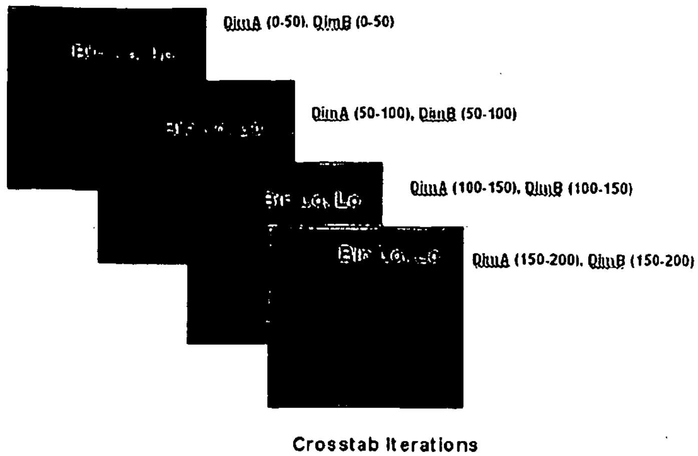
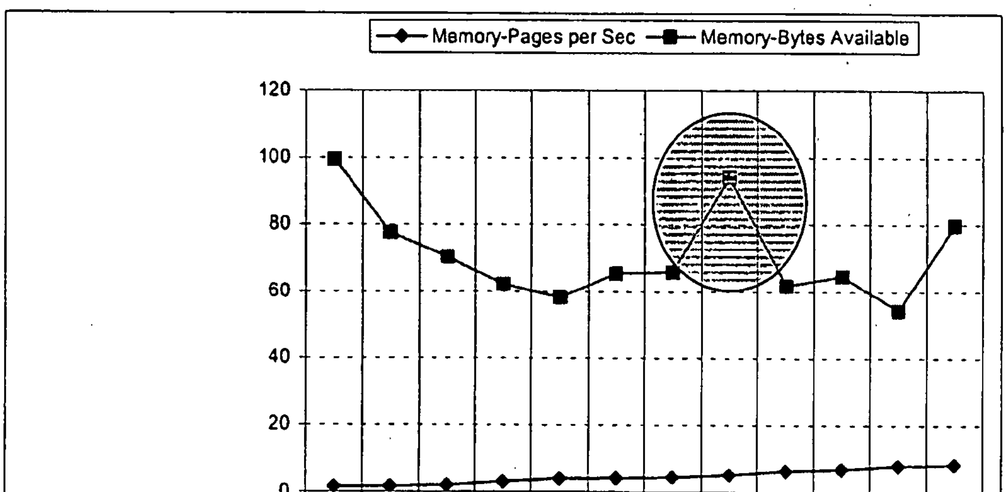


FIG. 19



	1	2	3	4	5	6	7	8	9	10	11	12
Memory-Pages per Sec	1.73	1.84	2.15	3.23	4.19	4.31	4.63	5.33	6.48	6.97	7.92	8.34
Memory-Bytes Available	99.63	77.57	70.42	62.25	58.27	65.42	65.89	94.24	61.73	64.64	54.3	79.77

FIG. 20

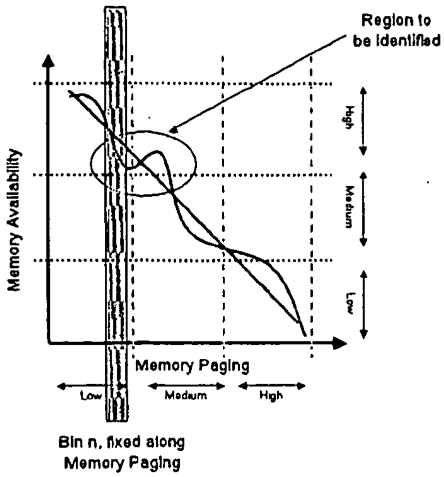


FIG. 21A

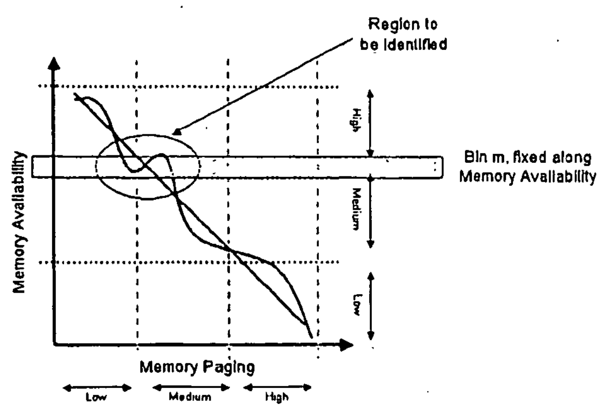


FIG. 21B

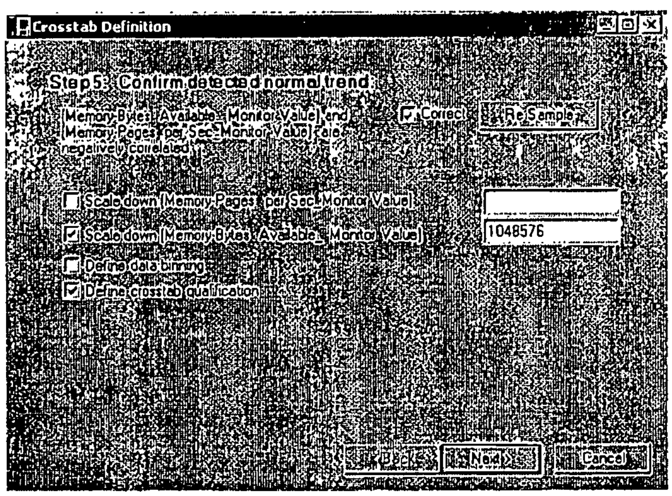


FIG. 22